April 2006

Observability of Separation Events Using Radar Data

Brandon Michael Allen
Worcester Polytechnic Institute

Carol Jeanette Carveth
Worcester Polytechnic Institute

Project Registration Number: PH-AZ-0601/PH-AZ-0602

Observability of Separation Events Using Radar Data

A Major Qualifying Project Report:

submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Bachelor of Science

by

Brandon M. Allen

and

Carol J. Carveth

Date: April 27, 2006

Approved:

Professor Alex Zozulya, Major Advisor

Abstract

This project examines the usefulness of radar data as displayed in a Range/Time/Intensity (RTI) plot. There is a focus on understanding what can be uniquely determined about a separation given information about later states of the system. Separation events were simulated from which RTI plots were created. Analysis determined that while determining a few parameters of the event was possible, the unique situation could not be rediscovered. Recommendations for future work include higher level models and longer time scales.

Authorship Page

Both group members contributed equally to the completion of both the project matter and the writing of this report.

Acknowledgements

We would like to thank the following people for their contributions to our MQP:

Dr. Stephen Weiner of MIT Lincoln Laboratory, whose support and encouragement were invaluable as our group's Liaison to Lincoln Laboratory.

Professor Alex Zozulya of Worcester Polytechnic Institute, our Advisor, whose guidance and understanding were well-appreciated.

Roger Sudbury, the MIT Lincoln Laboratory Liaison to WPI.

Dr. Robert O'Donnell, the former MIT Lincoln Laboratory Liaison to WPI.

Shelby Soleimani of MIT Lincoln Laboratory Human Resources, whose cheerful attitude and businesslike manner were very helpful to us.

Lisa Vaira-Haslock of MIT Lincoln Laboratory, whose help made this project easier.

Other Members of MIT Lincoln Laboratory for their friendly encouragement.

Table of Contents

List of Figures

List of Tables

Executive Summary

The goal of this project was to explore the information that can be gathered by interpreting Range/Time/Intensity (RTI) plots of radar data. To accomplish this goal we used a system of creating radar data for simulated situations. The situations our goal was focused on were events during which one object made up of a cone and a cylinder of equal radii joined at their circular bases separated into a cone and a cylinder by way of a separating impulse.

This project is part of a larger effort to be able to completely determine all parameters of such a separation event and to control the resultant pieces the instant the event takes place. Limiting factors that have, to this point, prevented this goal's attainment include an inability to accurately determine the precise moment of separation and the difficulty of distinguishing between the separated objects, among other factors. The method we were presented with as being the common method of interpreting radar data is the RTI plot, so we used this method in our study of the situation. Before we simulated the data and interpreted the results we investigated the workings and capabilities of commonly used radar systems, as well gaining as much knowledge about how to read an RTI plot as we could with the resources that we had access to during the course of the project.

Important information that we wished to discover was how different parameters of an separation could interact to result in RTI plots that were too similar to tell apart for times after the separation, as well as whether the separated pieces would knock together after separation and behave in a manner which could result in incorrect interpretations of the initial separation parameters. This meant that we also had to decide how we could determine if the two objects collided, so some research was done about how we could discover that.

Since simulating three-dimensional objects with three-dimensional motions with all real forces seemed an insurmountable task, we cut the task down significantly by making a number of assumptions and simplifying the problem. We assumed that the objects would behave as two-dimensional objects with rotational and linear motions. Another assumption we made was that there were no outside forces acting on the two objects. Additional assumptions included that there was no uncertainty in finding the locations of the points on the objects, that the times at which the locations were found were 0.10 seconds apart, that only the discontinuities in the surface reflected anything at all of the radar signal, and that each point that was able to reflect the signal reflected it with the same intensity. The assumptions having been made we then went about simulating the data. Upon having simulated a large number of trials we created range/time/intensity plots using the simulated data, and drew conclusions based on these simulations. We concluded that we were able to tell the maximum lengths, ratio of masses, and locations of centers of masses quite well from examining the data. We also concluded that we had not used a long enough time scale in some instances to have enough information to determine even the post-separation situation. Our recommendations for future work include using more robust models of the physical situations and using a longer time interval. We also recommend that future works incorporate information from other types of sensing systems.

1 Introduction

In almost all cases, a measurement with a low level of uncertainty means that the measurement is more useful than a measurement with a high level of uncertainty. There are bounds that exist for how low uncertainties can become, which can be developed from models of how the measuring devices are used. The purpose of this project was to determine some of the boundaries that exist in how much information can be gleaned from reading a specific representation of radar data.

Radar is a technology that is used to learn about the motions of distant objects. In most cases, the primary goal is to learn what the object is and where it is going. A radar system accomplishes this task by transmitting radio waves and receiving any reflections that it is able to. Attempts are then made to analyze the information gained from the frequency, amplitude, and direction of the received signal to find relative speeds, sizes, and shapes of the object or objects that were detected. One of the tasks of Lincoln Laboratory, which is located in Lexington, Massachusetts, is to investigate different problems that can be present in such a system, and then to find ways of minimizing those problems. The problem that we endeavored to learn more about was how much could be learned about a separation event, given information about later motions of the two pieces, where a separation event can be defined as a single object dividing into two objects. We chose to approach this problem by simulating perfectly measured data, to find out if the radar data, itself, is the cause for the higher level of uncertainty, rather than errors in the measurement-taking processes.

Our procedure for isolating the problems present in making conclusions for the data itself involved simulating the data without simulating the measurement errors to go with the data. We simulated over 50,000 trials of the separation of a two-dimensional triangle and a two-dimensional rectangle. These were meant to model three a three dimensional cone and a three dimensional cylinder, attached at the bases. We predicted which tracks we thought we probably could not tell apart, based on the differences and similarities of the conditions we were imposing on the separation event, and checked the results against each other to find out if changing one or two of the characteristics in our model of the separation changed the overall final motion. We picked a few of these files and checked them against each other. We also chose a few other files, just to compare to see if we could uniquely determine what the original interactions at the time of separation were, merely by looking at the tracks after the separation event.

We found that only one of the aspects of the separation that we had considered as possibly causing our failure to distinguish between tracks actually prevented us from noticing a difference between the tracks. There was also the rather surprising result that changing one of the variables in some instances shifted only one of the two objects' tracks, leaving the other object's motion relatively unchanged.

These results were somewhat surprising, as we had predicted that changing two of the variables inversely would have made the tracks indistinguishable from each other. We found that the tracks were so complex and that the information about the separation was so inter-related with itself that in each trial we were able to tell the plots of the two situations apart. The second part of our procedure, which involved attempting to find the relationship between the masses and the location of the force that caused the separation event, was less surprising. We found that, just as making one event look like another by changing only two or three of the inputs was nearly impossible due to the inter-relatedness of the inputs, it was impossible to determine exactly what the state of the object at separation was looking only at the final motions and positions.

Based on our findings that changing only a few of the inputs resulted in entirely unique results, we concluded that if at least a few of the aspects of the initial separation were known it was probable that the rest of the situation could be determined. The minimum number of already-known pieces of information that we determined was three. Our inability to determine exactly what the inputs were based on knowing the outputs confirmed this.

Since we only undertook the analysis of two-dimensional objects in two-dimensional motion, we would recommend further examination of three-dimensional objects with three-dimensional motions. We also recommend that the possibility of the initial object separating into more than two objects be considered. Additional possible aspects of the separation, such as external forces, could also be considered.

2 Background

In this project we examined the possible problems that could arise in the data analysis of radar data based upon radar data if there were no measurement errors in acquiring the data. We first needed to explore the workings of radar systems. To answer the question of whether the objects would collide, we needed to learn more about testing for collisions.

2.1 Radar Systems

There are many types of radar sensor systems, but they all have a few things in common. The word “radar” stands for “RAdio Detection And Ranging” (Edde, 1995). There is a great deal of information to be gathered from sending a signal from an antenna and receiving a reflection or scattering of that signal at either the same or a different antenna. Sensing that an object is present using radar only requires the radar to detect a reflection or scattering of the signal that was sent out. The velocity of an object can be found by comparing the frequency of the reflected signal to that of the sent signal and taking into any shift into account as a result of the Doppler Effect (Edde, 1995). Some of the motions about an object’s center of gravity can also be found as a result of the Doppler Effect, as if there is any rotation, the signal from the part of the object rotating away will be red-shifted, and the portion of the object rotating towards the antenna will blue-shifted, so that the shift is not uniform across the entire signal of the object. To discover the position of the object sensed, the known speed is used with the amount of time required for the signal to travel to the object and then to the sensing antenna to find the range from the antenna. The direction (angular position) from the sensing antenna is found by comparing the direction of the reflected signal to a reference signal. The shape and size of the object can be found from the phase of the signal and by comparing how the signal has been scattered back to the sensing antenna with the known scattering of known objects (Skolnik, 2001).

Radar devices have been being developed for over a century now, the first demonstrations of its possibilities for usefulness dating back to Heinrich Hertz's experiments between 1885 and 1888. The systems developed were able to sense objects, and in some cases they were able to provide information about an object's location. Further development of these technologies in these countries and others was encouraged and possibly accelerated by its demonstrated usefulness in World War II, as well as the Cold War after it (Skolnik, 2001).

Even though wars seem to have assisted in the development of radar, there are many uses for radar even in times of peace. These include, but are by no means limited to, such applications as law enforcement and air, land, and sea traffic control, detecting and measuring objects underground, mapping, and detecting and locating weather phenomena (Skolnik, 2001).

2.1.1 Operating Parameters

There are essentially three parts of a modern radar system: the transmitter, the receiver, and a computer. The computer controls the frequency and strength of the signal that the transmitter emits, and analyzes the signals that the receiver takes in from its surroundings. It is frequently the case that the receiver and the transmitter are part of the same instrument, but this requires special precautions, so the receiver will not become damaged (Wehner, 1987).

The process involved in radar detection involves the transmitter sending out a pulse or continuous signal at a set frequency and amplitude. This signal is directed or merely sent out in all spherical angles. When a portion of the signal encounters a reflective surface, that portion of the signal will reflect off of the surface following the laws of reflection. The object that reflects the signal from the transmitter is often referred to as the “target.” If the receiver is positioned in such a manner that the reflected radio signal reaches it and the signal’s power is strong enough to be recognized as a signal, the computer analyzes the signal to find the speed and distance away of the reflective object.

If the signal is assumed to suffer no losses when propagating through air or vacuum, then the amount of signal incident on the reflected object is the fraction of the total area over which the signal has spread at that distance from the transmitter. If no effort is made to shape the transmitted signal, then this will be the fraction of the surface area of the sphere around the transmitter at that radius that the reflecting object takes up. The portion of the reflected signal detected similarly depends on the portion of the area through which the reflected signal is traveling at the distance between the reflected signal and the receiver that the receiver takes up. The strength of the signal received is thus greatly reduced from the strength of the transmitted signal, unless the objected reflecting the signal is perfectly reflective and takes up the entire radial area that the signal travels through, and the receiver is lined up in such a way so that its area is sufficient to receive the entire signal. In general, if the transmitter and the receiver are at the same location, the signal strength is reduced by approximately a factor of the fourth power of the radius between the transmitter and the reflecting object by the time it returns to the receiver. This means that the receiver must be able to accept and differentiate between signals that are many times lower in strength than the transmitted signal.

The receiver collects the radar signals' frequencies and amplitudes when it is able to receive information, and only for given frequencies and amplitudes. Signals too strong (signals with amplitudes much higher than the expected signal strength) can overload the receiver's circuits. It is for this reason that in devices where the transmitter and receiver are very close to each other that the transmitter cannot be transmitting in a manner that it would be possible for the receiver to receive a direct blast from the transmitter.

The emitter sends out electromagnetic wave signals to its surroundings at a set wavelength and in a range of directions. These directions can be structured by a dish surrounding the emitter. Smaller ranges in directions at a given signal strength result in greater signal strength in those directions, resulting in greater distances away from the radar that can result in returned signals. The electromagnetic waves used for radar purposes are most frequently radio waves or microwaves.

The amplitude of a returned signal is dependent on the amplitude of the transmitted signal, the distance that the signal had to travel, and the reflectivity of the surface that reflected the signal. Objects can be made to have very low surface reflectivities for radio signals, thus making them less visible to radar systems. Discontinuities in the surface, such as edges, are frequently more reflective than the continuous parts of surfaces. For our simulations we assumed that only the reflected signals from the corners of the rectangle and the triangle had high enough amplitudes to be recognized as signals by the receiver. We also assumed that all reflection from the target were received, and that only signals were received, not noise. For a three-dimensional cylinder or cone the discontinuous edges would have been the most visible, which would mean outer edges of the circular bases of the cylinder and the cone, and the point of the cone.

It is acceptable to assume that the signal propagates through approximately the same solid angle very far away from the transmitter as it does directly next to the transmitter. It can also be assumed that the reflected signal causes the signal to spread out in a similar manner. This results in an inability of the receiver to precisely sense the direction of a signal, making the uncertainty in the angular direction quite large.

Radio waves are electromagnetic waves, and so travel at the speed of light as affected by different materials' indices of refraction. The primary materials that the waves propagate through are air and vacuum, both having indices of refraction of approximately 1, leaving the speed of a radar waves the generally accepted value of the speed of light. It is, therefore, possible to measure the distance that the signal traveled between leaving the transmitter and arriving at the receiver if the time between the transmittance and receiving of the signal is known. If both the transmitter and receiver are in the same location then the relative distance of the object reflecting the signal from and back to that location can be known. If, however, the distance between the transmitter and the receiver is not negligible then there will exist a relationship between the distances between the object reflecting the signal and the transmitter and the receiver. Technology has advanced to the point that it is not unusual to be able to very accurately tell how much time elapsed between the transmission of the signal and the receiving of the reflected beam. Since the speed of light in air and in vacuum has quite low uncertainty and the measured times of travel can have very low uncertainty the distance can be computed with very low uncertainty (Skolnik, 2001).

The speed of the target can be found with as few as one discrete signal's transmittance and return, if the wavelength of the signal sent and the wavelength of the received signal are known. It is a consequence of the non-relativistic Doppler Effect that the transmitted wavelength of a signal and the wavelength that the receiver gets of the same signal can be different. If the target reflecting the signal is radially stationary with respect to the transmitter-receiver system, then the signal will have the same wavelength at the time of transmittal as it does when it is later received. If the signal is moving closer to the transmitter-receiver system then the wavelength will be shorter, also known as being blue-shifted. If the target is moving away from the transmitter and receiver then the signal will be red-shifted. Since the wavelength of the radar signals is long, the very small shifts resulting from non-relativistic speeds are still discernable (Skolnik, 2001).

One particular use of radar is tracking radar. This makes use of many iterations sending a signal and receiving reflections from the target(s). The range(s) are kept track of, and plotted vs. time to demonstrate relative radial motions, among other things. One common method of plotting this data is a Range/Time/Intensity (RTI) Plot, which is discussed in section 2.1.3 of this paper.

2.1.2 Limitations

Radar systems have many limitations which hinder their usefulness. Limitations of primary concern seem to be limited range of usefulness, noise in the signal, decoys or other unintended reflections from non-targets, timing complications, and accurate resolution of the position of targets. The limited range of usefulness is dependent on the radar equation, which was mentioned earlier, as well as the curvature of the earth. A signal will diffuse enough over long enough distances that any detection of the reflected signal will be too low to be able to gain any useful information.

The limited range of usefulness for a given radar system is caused by the scattering and diffusing of the radar signal through the air or other media, as well as the spreading out of the signal. The farther the signal travels the worse these effects become. Small target cross-sections become worse problems at greater distances from the transmitter-receiver systems, as well. Small targets take up smaller proportions of the area of transmitted power, but the larger the radius of travel, the larger the surface area of the boundary of transmitted power (Wehner, 1987).

Noise in the received signal can have many sources. One of the possible sources is the circuitry that makes up the receiver, or even other nearby electronic equipment (Skolnik, 2001). Noise can also arise from scattering of the emitted signal, rather than only direct reflection, or other electromagnetic wave sources interfering with the signal of interest. Decoys and other unintended reflections also will confuse the signal. Decoys are actual targets that are merely not the target of interest. Due to the low-angular resolution capability that exists for most radar systems it is frequently useful to create a decoy that can become unresolved with a target of interest. A person attempting to track the target of interest is then forced to keep track of both the decoy and the target of interest after they become resolved again unless they are able to confirm in some way that the target of interest is one or the other of the freshly resolved objects.

Since many radar systems operate using a system of pulsing the transmitter and only receiving signals while the transmitter is off, several types of difficulties in analyzing the times of received signals, two of which will be mentioned in this paper. One difficulty is if the signal is returned during a period of transmittal. A correction for this can be to vary the time periods between each transmittal in a measured manner, but this can extend another difficulty. It is possible for targets to be far enough away that their reflection will not arrive at the receiver until after the next transmitted pulse is sent out. If a target appears after transmitted signals have being sent out for a long time, it is difficult to determine which transmittal is associated with that target in determining the amount of travel time for the signal.

The low-angular resolution capability of radar systems, as previously mentioned, can be hundreds of times worse than the radial resolution capabilities of targets. Shielded targets, however, can also be difficult to resolve in the radial direction (Wehner, 1987). Since angular resolution is so inaccurate, it is often not represented when plotting the rest of the information. This is true for Range/Time/Intensity plots.

2.1.3 RTI Plots

One of the methods used to interpret radar data is known as a Range/Time/Intensity (RTI) Plot. The plot compares three sets of data: a relative range of signals from the radar antenna, the relative time at which measurements are made, and the relative intensities of the signals received. Normally, the range is plotted along the horizontal axis, time along the vertical axis, and intensity with a color bar as shown in Figure 2.1. Figure 2.1 is the only proper RTI plot to which we were given access, but we were not told what it actually represented. We did not look into relative intensity differences as we had no way of modeling them accurately.

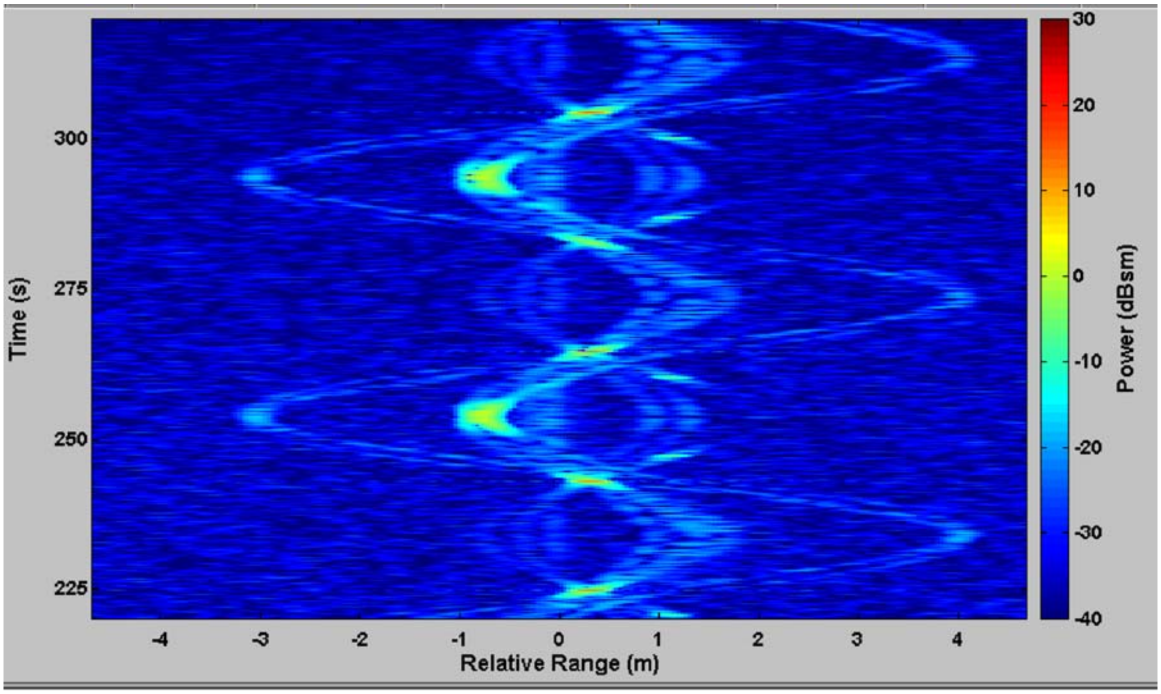


Figure 2.1: Example RTI Plot – Range/Time/Intensity Plots are frequently used to display information gathered using radar systems. The range axis is a radial range, relative to some set point. The time axis, in seconds, displays the time at which each of the signals was received. The intensity scale was determined by the strength of the received signals.

RTI plots can, at times, be difficult to interpret. Radar sensors in general are very good at measuring range with small uncertainty, but they have very large uncertainty in angular resolution. The paths on a RTI plot will often cross and connect, but that does not mean that the two objects being tracked are anywhere near each other, they could be miles apart. Merely, they are the same distance away from the sensor. We use Figure 2.1 as an example. In Figure 2.1, there are six sinusoidal paths and one linear which are all traced out in Figure 2.2 for clarity. The linear path (Path 1) is located at the relative range of zero meters indicating that there is an object, or part of an object, that does not move away or toward the radar antenna. Path 1 however is not continuous, indicating that the point is not always visible. Sinusoidal paths indicate object oscillation. With the six paths crossing each other every half-period, it is difficult to define the individual paths. It is clear that there are two separate paths with much larger amplitudes than the rest; one ranges between about -3m and 1-2m (Path 2) (The path seems to widen on the positive side of the zero range indicating that the object being tracked does not have a sharp corner, but is more of a curve) and the other from about -0.5m to 4m (Path 3). There are two smaller amplitude paths that have approximately the same amplitude but in opposite directions ranging from -0.33m to 1m (Path 4 being in the negative at time 230s and Path 5 being in the positive at the same time). There are then two slightly larger amplitude paths also opposite from each other, but not with the same amplitudes. One follows the same oscillation as Path 4, being at a negative range at 230s, and ranges from about -0.66m to 1.33m (Path 6). The last path (Path 7) is interesting because it is not a continuous path like all the others. Path 7 only shows in the positive range, and disappears at about 1.5 before reappearing at the same range twelve seconds later. It also disappears at every junction of paths. Of course it is also possible that Path 7 does indeed have a visible negative range section while Path 4 does not. Some things we do know are that the objects have a period of rotation of about 38s. Paths 4, 5, and 6 seem to rotate about a single range of about 0.33m which could indicate that they are all mapping different parts of the same object, whereas Path 2 appears to rotate about -2.25m and Path 3 about 2.75m.

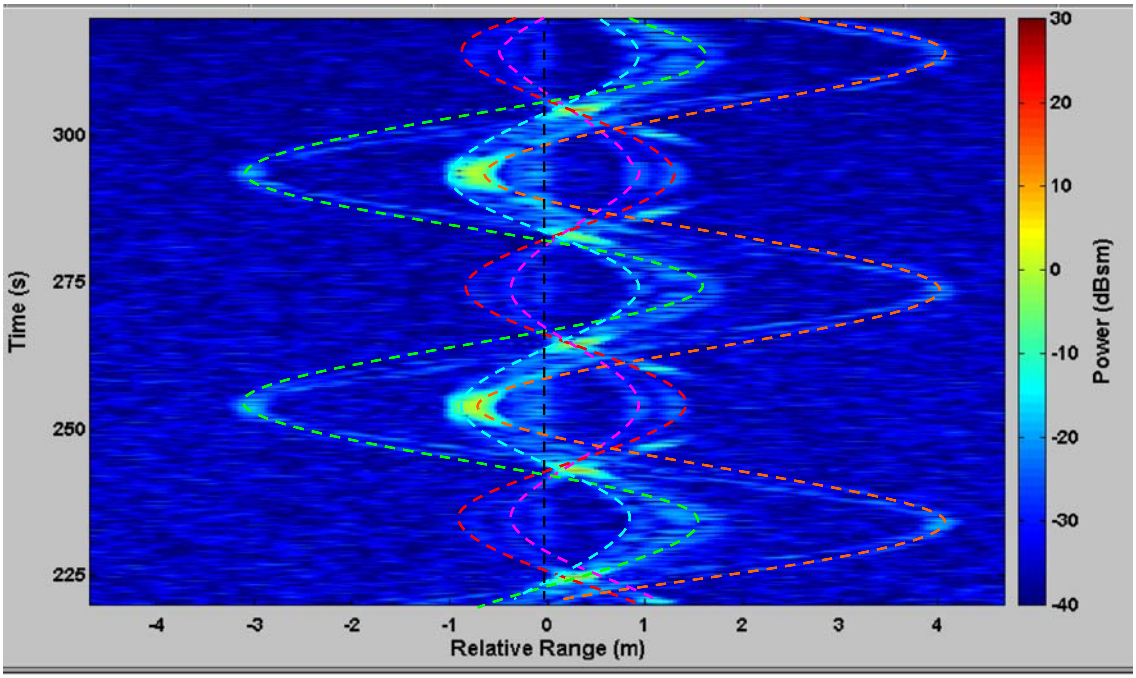


Figure 2.2: RTI Plot with highlighted tracks – This RTI plot displays the same information as the plot in Figure 2.1, but it includes dashed lines to highlight individual tracks.

2.2 Collision Tests

One cannot use information from a radar scan to determine the history of a moving object without having previously seen examples. Using examples in real life would be far too costly and impractical, so we had to use computer models of various possible object movements. One of the things we wanted to determine is whether or not two objects, originally starting as one, would collide with each other when they separate. To figure out how to do this, we searched through a book that discusses collisions. We found a book by Christer Ericson to be an excellent source of help. While the book focused on the use of real-time collision detection in video games, it provided us with some useful concepts. One suggestion was to place an object in a bounding volume to simplify the shape of the object when making collision tests. The first and simplest bounding volume is a sphere completely enclosing an object, centered on a specific point in the object. If the central points of two objects are ever at a distance closer than the sum of the radii of the spheres around those objects, then there is a chance that the objects collide. Other bounding volumes that had potential are axis-aligned bounding boxes (AABBs) and oriented bounding boxes (OBBs). An AABB is a box bounding an object that has the vectors normal to the faces of the box made parallel to the axes of the coordinate system used. An OBB is similar to an AABB, but the normal vectors are parallel to the axes of the object itself. The two boxes, while similar, have some important differences. The AABBs, since they are related to the coordinate system, will change size and shape as the object rotates. The OBBs on the other hand do not change shape, but rotate along with the object (Ericson, 2005).

Another useful concept was the separating-axis test. The separating axis theorem states that given two convex sets A and B, either the two sets are intersecting or there exists a line P such that A is on one side of P and B is on the other. A ‘separating axis’ is the line perpendicular to the line in the gap between the two objects. If the projections of the objects onto the separating axis overlap, then the objects overlap or collide. (Ericson, 2005) The trick is to find an actual separating axis. We found it useful to test axes that were perpendicular to the sides of each oriented bounding box containing the objects.

3 Methodology

Since this project was to examine whether it is possible to find information about an unresolved splitting of an object by looking at the pieces of the object after they have become resolved, we chose objects that we thought would be useful to know about: a cone separating from a cylinder. It was decided that looking at this problem in three dimensions would have been far too complex for this particular effort in resolving the problem so we approximated the cone and cylinder by a triangle and a rectangle respectively. (As this problem is being solved through a larger effort, later takes on this problem will include more robust models.) Since our model was only two dimensional, we ignored the effects of the objects spinning along their primary axes. Other assumptions made when setting up our models were a

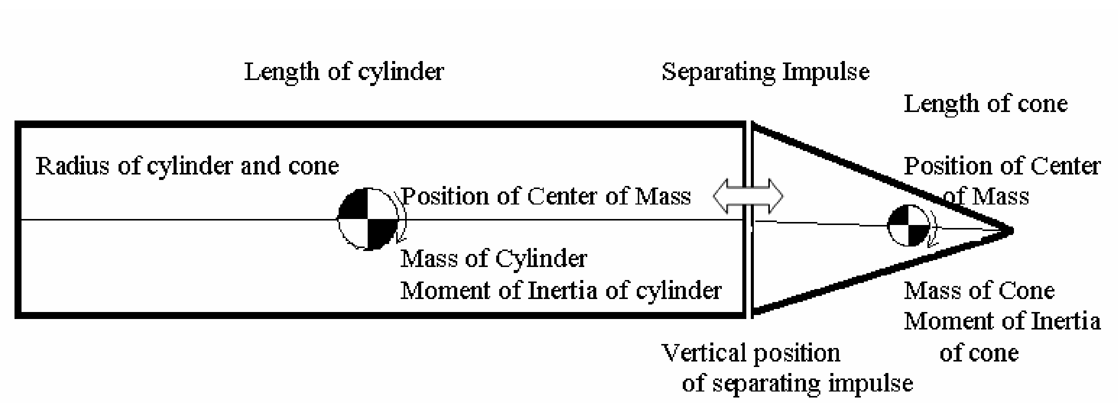


Figure 3.1: Reference Figure for System Considered - For each simulation we considered a system which looked something like the system pictured in this figure. A rectangle and a triangle were separated by an impulse at some distance above or below the central axis. Both of the individual centers of masses were located along the central axis, and the positive direction of rotation was assumed to be into the page. All of the mentioned aspects of the separation event were varied, and at least one was different between each simulation.

lack of air resistance or other outside forces, which allowed us to use conservation of momentum; radial symmetry in mass distribution, keeping the centers of mass on the primary axes of each object; and that we knew the moments of inertia of each object about their centers of mass. Our basic model was that of a rectangle and a triangle, with equal base radii, having no initial rotation about the combined center of mass and no initial separation between the bases of the objects, separating after experiencing an internal repulsive force along the primary axes for a short period of time as shown in Figure 3.1.

3.1 Simulation of Data

We wanted to see what would occur if certain variables were changed. We changed the physical dimensions of the objects by varying the length of each and the shared radius. We changed the mass distribution with differences in the ratio of masses, the moments of inertia, and the positions of the individual centers of mass relative to the initially shared base. We also used variations of the separating impulse by changing the magnitude of the force and the distance from the initial primary axis. (The time over which the force was applied we kept constant.)

When modeling the situations, we first had to calculate the linear and angular velocities of both objects resulting from the separating impulse acting on the objects. This was accomplished using basic laws of equal and opposite forces and conservation of momentum. We then tested for collisions.

To determine whether or not a collision occurred in our models, we first used spheres centered on the objects centers of mass as bounding volumes. Since the two objects are separating, it was apparent that if the distance between the two centers of mass was greater than the sum of the radii of the bounding spheres then the objects could never collide. During the time it took for the objects to separate to this point, it was necessary to check a little closer for collisions. We decided that using the separating-axis theorem would work well using modified oriented bounding boxes. The cylinder could be approximated by a rectangle, and the cone as a triangle. After testing the possibility of collision with the bounding spheres, we went on to test axes perpendicular to the sides of the objects. Each of these tests would have had the edge of one object oriented vertically. If the other object projected onto the possible separating-axis farther from the center of mass of the object with a vertical edge than the edge itself then the possible separating-axis was an actual separating-axis. If all of the edges of the two objects were perpendicular to a separating-axis, then there was no collision.

If there was a collision, we did not go any further with that particular trial as we wanted to focus on simple motion and not get into the more complex motions resulting from rebounding. For those trials in which no collision occurred, we used the initial conditions and the resulting equations of motion to find the position of each corner of both objects as they moved through time. This was done with simple trigonometry. Plotting this data directly would have made as useful RTI plot, but not a realistic one. In the process of rotating, the objects tend to obscure the line of view of their own corners and that of the other object. Again, using some trigonometry allowed us to determine at which times a particular corner was ‘visible’ to the ‘sensor’ and at which times it was not. Using this data, we were able to make RTI plots (without intensity data) for each non-colliding trial. (Unfortunately we were unable to show the data points for the visible corners without showing the invisible ones as well, so we added a visibility factor that set invisible points to have a range of zero as can be seen in Figure 3.2.)

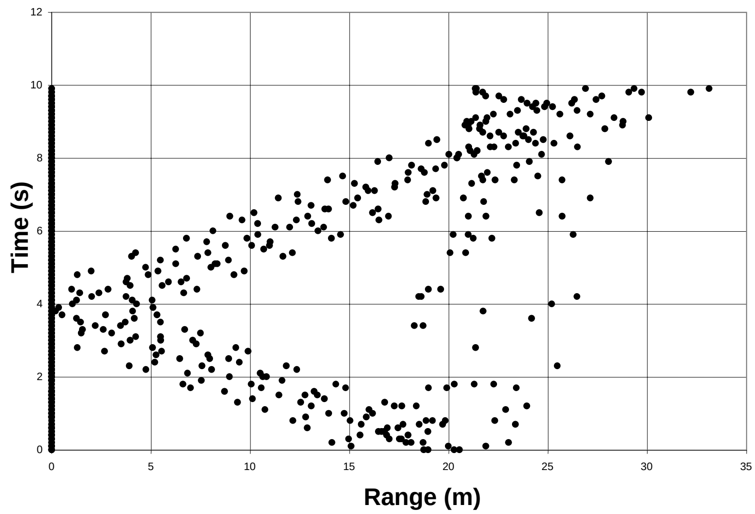


Figure 3.2 - Example of RTI Plot of Simulated Data – Computer software was utilized to create an RTI plot for several of the sets of simulated data. Range is plotted on the horizontal axis in units of meters, and time is plotted on the vertical axis in units of seconds. A location for each of the outer points was generated every tenth of a second, but all points that were behind other parts of the shapes from a set direction were considered to be "invisible", and are displayed as having a range of zero. Since the range has been plotted as an absolute value, what looks like a collision at t = 8s really is not a collision. Intensity is not plotted in this graph or in any other of the RTI plots of simulated data.

3.2 Process of Analysis of Simulated Data

Once we had our RTI plots of the various trials, we had to find trends in the changes of the plots. We decided that a good way of discovering the trends would be to make predictions as to which changes in variables we thought would result in an identical RTI plot. We then selected examples of each of the possibly confounding data and compared them to each other. We decided to look at three changes, observing these changes with a selection of different starting conditions. We considered the radius of the separating impulse and thought that having it in a negative position would result in the same RTI plot. We also looked at having the same separating torque but with different force and radius components. We also considered the possibility of the mass and moment of inertia ratios and the different impulse strengths at different positions causing similar enough RTI plots where we wouldn't be able to tell them apart. We considered the simplicity of multiples of all of the variables, but decided that this would be irrelevant to show, due to the obviousness of the problem, and did not include in our simulations. We chose four sets of data at random, created an RTI for that data, and compared them to (1) the RTI that had the same inputs but with an oppositely signed radius of the separation impulse, (2) the RTI that had the same inputs but with a larger force at a smaller radius, and (3) the RTI that had the same inputs but with a different relationship between the three factors of separating impulse, position of separating impulse, and mass ratio. After comparing these sets of RTI plots, we made a few conclusions about the interaction of those three sets of input variables.

Having come to a few conclusions on what effects in the behavior of a separation event come about by changing certain physical variables, we wanted to see if we could use these conclusions to work backwards as is the goal of this experiment. We selected three random RTI plots, without initially looking to see what the inputs were in those cases, and attempted to derive the conditions of the separations. After viewing our calculations, we then checked them against the actual data to compare.

4 Results

Our predictions of possible confounding across variables were as stated in the previous section. We found that, fortunately, in most cases the two objects did not collide after separating, meaning that in most cases if we were able to find a good extrapolation of what was happening at the time of separation, no inaccuracy would result because of collision effects. We used the initial separation event parameters as shown in Table 4.1 to analyze whether or not the interaction of the input variables could make two or more of each set of four RTI plots appear to be the same. The trials 1a-1d were varied by changing the location of the separating impulse. The trials 2a-2d were varied by changing the separating impulse, the location of the separating impulse, and the mass of one of the objects, or, in effect, varying the internal torque of the system. Trials 3a-3d were varied by changing the separating impulse and the mass of one of the objects. The parameters of trials 4a-4d were varied by changing the mass ratio, moment of inertia, separating impulses, and position of the separating impulse. The method for deciding which parameters to vary was iterative, and could conceivably be continued for every conceivable combination of parameters, but this would take a far longer time than would be considered appropriate. All sixteen of the RTI plots generated from these sixteen sets of initial parameters are shown in Appendix F.

The overall results of the examination of the RTI plots were that for three of the four situations involving only the change of the sign of position of the separation impulse, the RTI plots were exactly the same. The input variables were, actually, exactly the same in one instance, since the position of the separation impulse was centered along the line connecting the centers of mass of the objects. In the fourth case, when the RTI plot was not exactly the same, the plots differed by only two points on the graph, which both occurred at the same time.

	M_1 (kg)	I_{1o} (kg*m^2)	Center of $Mass_1$ (m)	M_2 (kg)	I_{2o} (kg*m^2)	Center of $Mass_2$ (m)	|F| (N)	t_F (s)	r_F (m)	L_{Cyl} (m)	R (m)	L_{Cone} (m)
Trial1a	1	0.45	0.4	0.1	0.04	0.85	50	0.01	-3	1	4	1
Trial1b	1	0.45	0.4	0.1	0.04	0.85	50	0.01	3	1	4	1
Trial1c	1	0.45	0.4	0.1	0.04	0.85	75	0.01	-2	1	4	1
Trial1d	1	0.45	0.4	0.5	0.2	0.85	250	0.01	-3	1	4	1
Trial2a	1	0.35	0.4	2	0.4	1.5	12	0.01	-1	3	2	2
Trial2b	1	0.35	0.4	2	0.4	1.5	12	0.01	1	3	2	2
Trial2c	1	0.35	0.4	2	0.4	1.5	6	0.01	-2	3	2	2
Trial2d	1	0.35	0.4	1	0.2	1.5	6	0.01	-1	3	2	2
Trial3a	1	0.3	1	0.7	0.4	2.55	20	0.01	0	2	0.5	3
Trial3b	1	0.3	1	0.7	0.4	2.55	20	0.01	0	2	0.5	3
Trial3c	1	0.3	1	0.7	0.4	2.55	5	0.01	0	2	0.5	3
Trial3d	1	0.3	1	2.1	1.2	2.55	60	0.01	0	2	0.5	3
Trial4a	1	0.4	3	5	0.4	0.5	5	0.01	0.5	5	3	1
Trial4b	1	0.4	3	5	0.4	0.5	5	0.01	-0.5	5	3	1
Trial4c	1	0.4	3	5	0.4	0.5	1.25	0.01	2	5	3	1
Trial4d	1	0.4	3	7.5	0.6	0.5	7.5	0.01	0.5	5	3	1

Table 4.1: Forced Values Defining the Separation Events - The values displayed in this table describe the initial conditions for each of sixteen of the simulations that we carried out. We compared four sets of four of these situations to determine whether they yielded incredibly similar or dissimilar radar tracks at times after the separation. The trials were compared within the other trials with the same number index by examining the RTI plots of each. Object 1, corresponding to subscripts of 1, is the rectangle (cylinder) while Object 2 is the triangle (cone), as shown in Figure 3.1.

When examining the consequences of changing the components of the total torque of the separating impulse, we saw that even though the total torque remained constant, the slopes of the plots change significantly, indicating that the linear momentum was affected, even while the angular momentum was not.

Changing the mass ratio and moment of inertia interestingly kept some of the paths on the RTI the same, while others were shifted. While this isn't precisely the degree of interaction that we had anticipated, which would have been that all plots were the same, it was an interesting result. The speeds between the two objects were apparently increased, though from the data we could not be sure if this was only linearly or also rotationally.

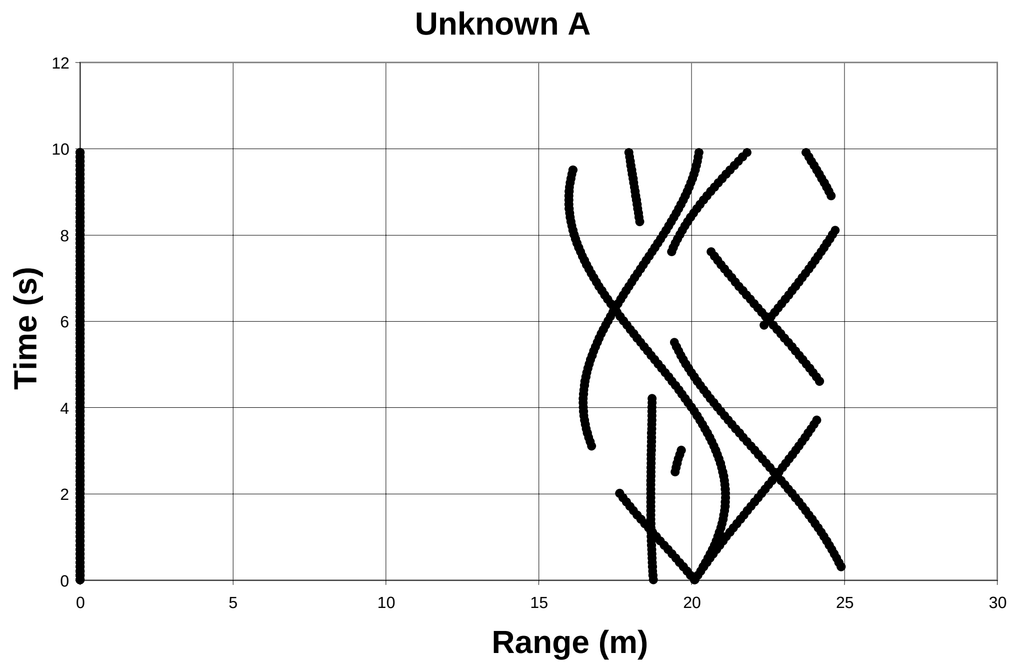
After we had analyzed several different radar tracks while knowing the separation characteristics, we attempted to arrive at conclusions about the separation event based on only RTI plots by themselves. The three we are using for our example in this context are displayed in Figure 4.1a-c. The simulated separation characteristics for each of these graphs are shown in Table 4.2.

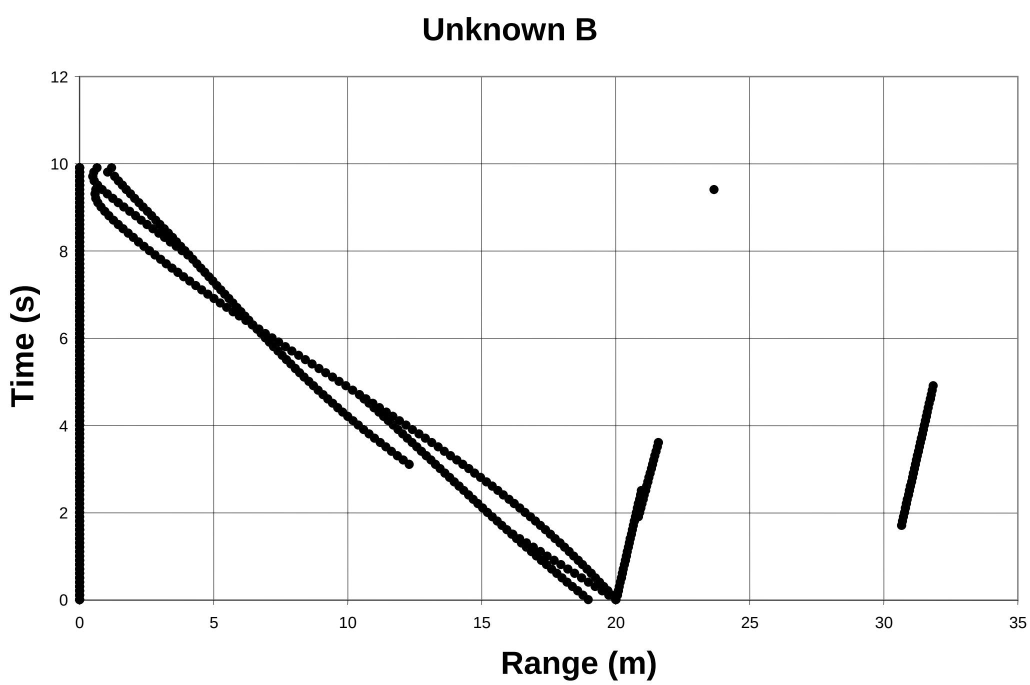
For the first RTI plot we were able to estimate the dimensions of the triangle and the rectangle. The estimate we made was that the greatest dimension of the triangle was 5, and the greatest dimension of the rectangle was approximately 1.5. An estimate was also made that I_1/I_2 was approximately 16/18.

	M_1 (kg)	I_{1o} (kg*m^2)	Center of Mass$_1$ (m)	M_2 (kg)	I_{2o} (kg*m^2)	Center of Mass$_2$ (m)	|F| (N)	t_F (s)	r_F (m)	L_{Cyl} (m)	R (m)	L_{Cone} (m)
Unk. A	1	0.4	2	1	0.3	1	10	0.01	-1.5	5	2	1.3
Unk. B	1	8.3	5	0.2	0.4	0.5	40	0.01	-0.5	10	1	1
Unk. C	1	4	5	0.01	0.5	1.5	10	0.01	0.25	8	2	3

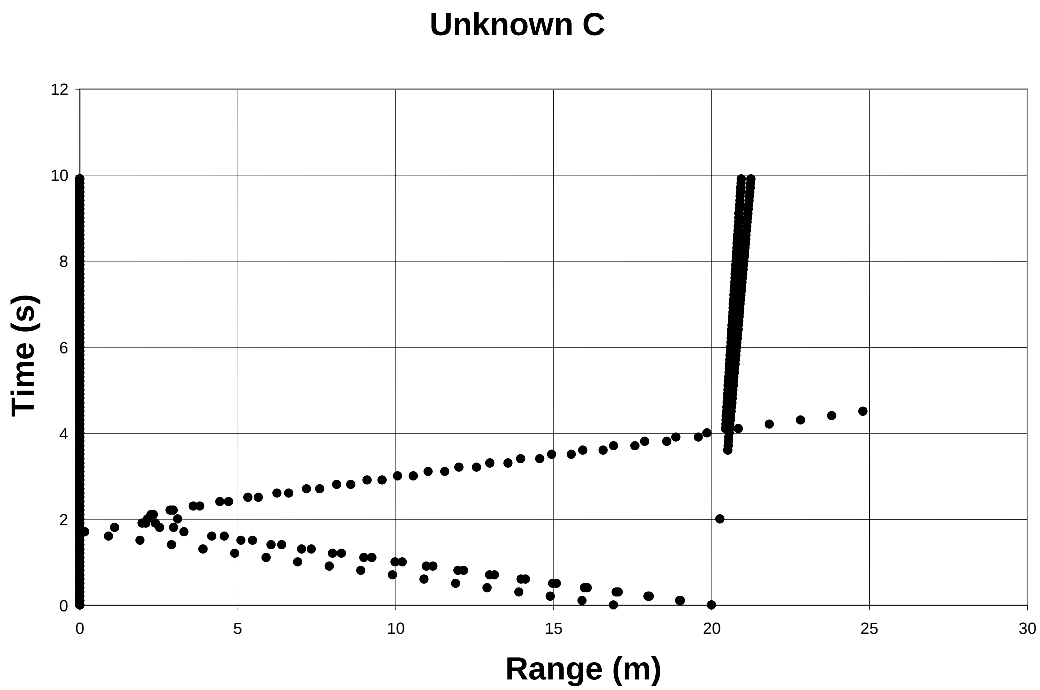
Table 4.2: Separation conditions corresponding to data in Figure 4.1 – We examined the generated RTI plots shown in Figure 4.1 and attempted to analyze the data to uniquely figure out the initial conditions for each RTI plot, for which the actual data is given in this table. "Unk." Stands for "Unknown"

Our second course of analysis included the determination that F was approximately equal to 12.5, the largest dimension of the triangle was 2, and that the center of mass of the triangle was at approximately 0.6. The length of the rectangle was found to be approximately 10, and the ratio of the masses was predicted to be m_1:m_2 = 5:1.


Unknown A
Time (s)
Range (m)
0
2
4
6
8
10
12
0
5
10
15
20
25
30


Unknown B
Time (s)
Range (m)
0
2
4
6
8
10
12
0
5
10
15
20
25
30
35

a b



c

Figure 4.1 (a, b, c): RTI Plots With Unknown Separation Characteristics – The graphs of Unknown A, Unknown B, and Unknown C, all with range units of meters and time units of seconds, were all examined without the examiner already knowing what the separation parameters were. The examination was directed towards figuring out as much about the separation parameters as was possible.

The third round of analysis resulted in a prediction that the length of the triangle was 3, and that $m_1:m_2$ = 18:1. We found this set of tracks particularly difficult to analyze, due to the odd nature of the motion of the tracks with the greater range.

Our method of determining which object was the triangle (cone), and which object was the rectangle (cylinder), involved determining which tracks were associated with each other. We knew that there could be a maximum number of three continuous tracks associated with the triangle, and a maximum number of four continuous tracks associated with the rectangle. In situations that there was very little rotation, it was easy to determine which tracks were associated with which other racks. In situations involving quite a bit of rotation about each of the two centers of the masses, this was more difficult.

Our overall result was that we were somewhat lacking in the ability to determine, uniquely, what the separation characteristics had been based on the RTI plot of the situation after the time of separation.

5 Conclusions

Our results indicate that there are very few different combinations of physical parameters that one would not be able to tell apart when studying the RTI Plot. We also found, unfortunately, that just being able to tell two tracks apart did not mean that we could describe all of the aspects of the separation event. We found that we were able to tell the maximum lengths, ratio of masses, and locations of centers of masses quite well from examining the data. We discovered, however, that we had failed to use a long enough time scale in some instances to have enough information to determine even the post-separation situation. The RTI plot is quite a useful tool, if some additional information is given then it is reasonable to suppose that almost all information about a separation event can be determined, but if only the RTI plot is presented, then some aspects, such as the direction of rotation, cannot be determined at all.

6 Recommendations for Future Work

The process of learning how to read radar data is iterative. One must start out, as we did, with a simple model. As one comes to an understanding of the features of a model, one must then expand the model to something higher. For instance, one of the limitations present in our model was the assumption that we had only two-dimensional translational and rotational movement. As we do not live in a two-dimensional world, it would be highly recommended to continue the process we started using a three-dimensional model. Also, there are many unresolved separation events that could occur, beyond the very basic separation event of a cone and a cylinder splitting apart. Objects can split apart several times over a period of time, or break apart into many pieces all at once. Modeling such events, while highly complex, would give a much more complete view of radar capabilities. Further study should go into the effects of outside forces on a separating system (i.e. gravity and air resistance). Gravity could, for instance, start the precession of an object's axis of rotation. We would also encourage the use of greater numbers of points in the generation of future simulations, if simulations are attempted. Combining the information of RTI plots with other methods of displaying data could also help limit the amount of uncertainty surrounding separation events. The work that can be done in exploring this area is far from completed, but we felt that this project accomplished at least a small part towards understanding some of the ways in which we can and cannot analyze radar data.

Appendix A – About MIT Lincoln Laboratory

In 1951, the Massachusetts Institute of Technology created the Lincoln Laboratory primarily as federally funded research center. Initially focusing on air defense technologies, they have expanded research into areas such as communications, space and tactical surveillance, missile defense, and air traffic control. They consistently follow a project from conception to simulation to design to demonstration.

Lincoln Laboratory employees developed the early ground-based radar systems and have consistently been at the forefront of radar technology development. They have been the ones that made significant contributions to radar technology including the Terminal Doppler Weather Radar, the Traffic Collision Avoidance System, and a new beacon surveillance system being implemented by the Federal Aviation Administration (FAA).

While work at the Labs is divided into several divisions with smaller groups in each division that have separate projects, there are many projects for which it is necessary to combine the resources of multiple groups to complete.

Appendix B – Run.m

```
%Run.m
%WPI Physics MQP A-term 2005
%MIT Lincoln Laboratory

%Run.m is a script which runs the functions
%{determine.m,Collision.m,RadarConvert.m} and saves the outputs to text files

%set up initial values

m1 1;            %Mass of object on left (cylinder)
m2=.1;           %Mass of object on right (cone)
I1=.35;          %Moment of Inertia of left object about left object's center of mass (cylinder)
I2=.01;          %Moment of Inertia of right object about right object's center of mass (cone)
cm1=.4;          %Distance from the central bases to the center of mass of the object on the left
cm2=.5;         %Distance from the central bases to the center of mass of the object on the right
f=26;            %Average Separating Force
r_f=-.07;          %Distance above the central axis at which the separating force is applied
t_f=0.01;        %Length of time that the separating force is applied...constant
cylength=1;      %Length of the cylinder (object on left/object 1)
rad=.1;          %Radius of both objects
conlength=1;     %Length of cone (object on right/object 2)
v1=0;            %Velocity of object 1 wrt Joint center of mass (cylinder)
v2=0;            %Velocity of object 2 wrt Joint center of mass (cone
w1=0;            %output of determine
w2=0;            %output of determine
dist 0;          %One of the ouputs of determine
N=1;             %Iteration count
t-0;             %incremental time
time 0;          %output of Collision Function holder
stop1=0;stop2=0;stop3=0;stop4=0;stop5=0;stop6=0;stop7 = 0; stop8 = 0; stop9 = 0; stop10 = 0;

%nested while loops

%Initialize inresults.txt, and fill in what each column represents
fid= fopen('3results.txt' ,'w');
fprintf(fid,'trial m1 I1 cm1 m2 I2 cm2 f t_f r_f cylL rad conL dist v1 v2 w1 w2 time\n');
fclose(fid);

while (stop1  == 0) %m2
    while (stop2  == 0) %f
        while (stop3  == 0) %rad
            while (stop4  == 0) %r_f
                while (stop5  == 0) %cylength
                    while(stop6  == 0) %conlength
                        while(stop7  == 0) %cm1
                            while(stop8  == 0) %cm2
                                while(stop9  == 0) %I2
                                    while(stop10  == 0) %I1

```

```
                %gives stuff
                    fid  = fopen('3results.txt' ,'a');
                    fprintf(fid ,'%g %g %g %g %g %g %g %g %g %g %g %g %g ',N,m1,I1,cm1,m2,I2,cm2,f,t_f,r_f,cylength,rad,conlength) ;
                    [v1 ,v2,w1,w2,dist]=determine(m1 ,I1,cm1,m2,I2,cm2,f,t_f,r_f);
                    fprintf(fid ,'%.12f %.12f %.12f %.12f %.12f ',dist,v1,v2,w1,w2);
                    time  = Collision(v1 ,v2,w1,w2,cylength,rad,conlength,(conlength-cm2),(dist));
                    fprintf(fid ,'%f\n',time);
                    fclose(fid) ;
                    Conepointlength =(conlength-cm2);
                %this is the tracking portion
                    t =0.01;
                    s = ['3trial' int2str(N) '.txt'] ;
                    fid  = fopen(s,'a');
                    if (time  == 0)
                        while (t  <= 10)
                            X  = -20;
                            Y  = 0;
                            RC  = RadarConvert(v1 , v2, w1,w2, cylength, rad, conlength, Conepointlength , cm1, t, X, Y);
                            fprintf(fid , '%12f %12f %3d %12f %3d %12f %3d %12f %3d %12f %3d %12f %3d %12f %3d\n' ,RC);
                            t  = t + .1;
                        end
                    else
                       while (t  <= time)
                            X  = -20;
                            Y  = 0;
                            RC  = RadarConvert(v1 , v2, w1,w2, cylength, rad, conlength, Conepointlength , cm1, t, X, Y);
                            fprintf(fid , '%12f %12f %3d %12f %3d %12f %3d %12f %3d %12f %3d %12f %3d %12f %3d\n' ,RC);
                            t  = t + .1;
                       end
                    end
                    fclose(fid) ;
                    N  = N+1;
                    t  = 0.01;
                    % %these are the closers on the loops:

                    if (I1  >= m1*cylength*cylength*.45)
                        stop10  = 1;
                    else
                        I1  = I1 + m1*cylength*cylength*.1;
                    end
                end % end of stop10
                stop10  = 0;
                I1  = m1*cylength*cylength*.35;
                if (I2  >= m2*(conlength*conlength)*.4)
                stop9  = 1;
                else
                I2  = I2 + m2*(conlength*conlength)*.1;
                end
```

```
                        end %stop9 end
                        stop9  = 0;
                        I2  = m2*(conlength*conlength)*.3;
                        if (cm2  >= conlength*.9)
                            stop8  = 1;
                        else
                            cm2  = cm2 + conlength*.2;
                        end
                    end %stop8 end
                    stop8  = 0;
                    cm2  = (conlength*.5);
                    if (cm1  >= (cylength*.6))
                        stop7  1;
                    else
                        cm1  = cm1 + cylength*.1;
                    end
                end %stop7 end
                stop7  = 0;
                cm1  = (cylength*.4);
                if (conlength >=5)
                    stop6  = 1;
                else
                conlength  = conlength+2;
                end
            end %stop6 end
            conlength =1;
            I2  = m2*(conlength*conlength)*.3;
            cm2  = (conlength*.5);
            stop6 =0;
            if (cylength >=5)
                stop5 =1;
            else
                cylength =(cylength+2);
            end
        end %stop5 end
        stop5  = 0;
        cylength  = 1;
        I1  = m1*cylength*cylength*.35;
        cm1  = (cylength*.4);
        if (r_f  >= .7*rad)
            stop4  = 1;
        else
            r_f  = r_f + rad*.7;
        end
    end %stop4 end
    stop4  = 0;
    r_f  = -.7*rad;
    if (rad  >= 5)%adjusts radius
        stop3  = 0;
    else
        rad  = rad + 2;
    end
end %stop3 end
stop3  = 0;
rad =1;
r_f  = (-0.7)*rad;
```

```
        if (f  <= 10)%adjusts f
            if (f  <= 1)
                stop2  = 1;
            end
            f  = f - 1.5;
        else
            f  = f - 12;
        end
    %range of forces, making range of impulses
    end %stop2 end
    stop2  = 0; %reset it for the next cycle
    f  = 26;
    if (m2  >= 5)
        stop1  = 1;
    else    %adjusts m2
        if (m2  <1)
            m2  = m2 + .4;
        else
            m2  = m2 + 2;
        end
    end
    I2  = m2*(conlength*conlength)*.35;
   %outermost level.  adjusts mass ratio.
end
```

Appendix C – Determine.m

```
function [vsep1,vsep2,wc1,wc2,s1] = determine (m1,I1,cm1,m2,I2,cm2,f,t_f,r_f)
%the function determine uses the masses of the two objects, the moment of
%inertia of both objects about their own centers of mass, the force
%applied (equally and oppositely), the amount of time the force was
%applied for, the distance above the central axis at which the force is
%applied, and initial angular velocity as inputs to provide the separating
%velocities and the angular momentum of the objects.

%This is assumed to be a 2 dimensional object with motion in the plane,
%object 1 being located to the left, object 2 on the right, and w is
%positive if spinning out of the paper, v's are positive to the right

m_t=(m1*cm1-m2*cm2)/(m1+m2);
%Position of the total center of mass...if it's Positive, it is in the
%first object.  If it is negative, it is in the second object.

%determine distances from individual centers of masses to center of mass
if m_t == 0
    s1 = cm1;
    s2 = cm2;
else
    s1 = (cm1-m_t);
    s2 = (cm2+m_t);
end

%compute torque shift (if there is one), based on r_f, t_f, f
tshift = r_f*f;
lchange = tshift*t_f;

%compute speeds of separation
vsep1=-f*t_f/m1;
vsep2=f*t_f/m2;

%compute angular velocities
%movement about system center of mass is determined elsewhere
%movement around individual center of masses
wc1 = lchange/I1;
wc2 = -lchange/I2;
```

Appendix D – Collision.m

```
%Collision.m
%WPI Physics MQP A-Term 2005
%MIT Lincoln Laboratory

function [timeCollide]  = collision(CylCent , ConeCent, CylRot, ConeRot, CylLength, CylRadius, ConeLength, ConePointLength , CylBaseALength) ;
%collision() takes in physical parameters of a separating cone and
%cylinder, the linear and angular velocities and dimensions and calculates
%whether or not the objects collide.

t = 0;                  %Time
%CylCent               Velocity of cylinder
%ConeCent              Velocity of cone
%CylRot                Angular velocity of cylinder
%ConeRot               Angular velocity of cone
%CylLength             Length of cylinder
%CylBaseALength        Length from cylinder Base A to center of mass
%CylRadius             Radius of cylinder and cone base
%ConeLength            Length of cone
%ConePointLength       Distance from cone point to center of mass
CylBaseBLength  = CylLength  - CylBaseALength ;
    %Distance from cylinder base B to center of mass
ConeBaseLength  = ConeLength  - ConePointLength ;
    %Distance from cone base to center of mass

%Spheres containing objects, centered at their centers of mass
if (CylBaseALength  >= CylBaseBLength)
    CylSphrRad  = sqrt(CylBaseALength ^2 + CylRadius ^2);
else
    CylSphrRad  = sqrt(CylBaseBLength ^2 + CylRadius ^2);
end
if (ConePointLength ^2 >= CylRadius ^2 + ConeBaseLength ^2)
    ConeSphrRad  = ConePointLength ;
else
    ConeSphrRad  = sqrt(CylRadius ^2 + ConeBaseLength ^2);
end

SeparateDistance  = ConeSphrRad  + CylSphrRad ;
    %Distance at which point objects cannot collide
TimeToSeparate  = (SeparateDistance  -  CylBaseALength  - ConeBaseLength ) / (ConeCent  - CylCent) ;
    %Time it takes to reach SeparateDistance
NumberIncrements  = 1000;
TimeIncrement  = TimeToSeparate  / NumberIncrements ;
DoesCollide  = false;
t = 0.00000000001;

while (t  <= (TimeToSeparate  + TimeIncrement))  & (DoesCollide  == false)
    CylPosition  = CylCent  * t - CylBaseALength ;
    ConePosition  = ConeCent  * t + ConeBaseLength ;
    CylAngle  = CylRot  * t;
    ConeAngle  = ConeRot  * t;

    %   B1 ____________ A1   C1
    %     |     b      |      |\ c
    %     |           a|    e |  > P
```

```
%      |_____________|       |/ d
%   B2               A2   C2

%X positions
CylXA    = CylPosition + CylBaseALength * cos(CylAngle);
CylXA1   = CylXA - CylRadius * sin(CylAngle);
CylXA2   = CylXA + CylRadius * sin(CylAngle);
CylXB    = CylPosition - CylBaseBLength * cos(CylAngle);
CylXB1   = CylXB - CylRadius * sin(CylAngle);
CylXB2   = CylXB + CylRadius * sin(CylAngle);
ConeXP   = ConePosition + ConePointLength * cos(ConeAngle);
ConeXC   = ConePosition - ConeBaseLength * cos(ConeAngle);
ConeXC1 = ConeXC + CylRadius * sin(ConeAngle);
ConeXC2 = ConeXC - CylRadius * sin(ConeAngle);
CylMaxXA = max(CylXA1, CylXA2);
CylMaxXB = max(CylXB1, CylXB2);
CylMaxX = max(CylMaxXA, CylMaxXB);
ConeMinXC = min(ConeXC1, ConeXC2);
ConeMinX = min(ConeXP, ConeMinXC);

%Y positions
CylYA    = CylBaseALength * sin(CylAngle);
CylYA1   = CylYA + CylRadius * cos(CylAngle);
CylYA2   = CylYA - CylRadius * cos(CylAngle);
CylYB    = -1 * CylBaseBLength * sin(CylAngle);
CylYB1   = CylYB + CylRadius * cos(CylAngle);
CylYB2   = CylYB - CylRadius * cos(CylAngle);
ConeYP   = ConePointLength * sin(ConeAngle);
ConeYC   = -1 * ConeBaseLength * sin(ConeAngle);
ConeYC1 = ConeYC + CylRadius * cos(ConeAngle);
ConeYC2 = ConeYC - CylRadius * cos(ConeAngle);

%'m' followed by two point names is the slope of the line connecting
%the two points in degrees (ex. mA1A2 is the slope of the line
%connecting CylA1 and CylA2.)
%'Im' followed by two point names is the slope of the line
%perpendicular to the same line.  (ex. ImA1A2 is the slope of the line
%perpendicular to the line connecting CylA1 and CylA2.

if CylMaxX < ConeMinX       %'shadows' on X axis don't overlap
     DoesCollide = false;
else                         %'shadows' on X axis do overlap
    mA1A2  = atand((CylYA1 - CylYA2) / (CylXA1 - CylXA2));
    ImA1A2 = mA1A2 + 90;     %axis perpendicular to a
    aCylXA1   = CylYA1 * sin(ImA1A2) + CylXA1 * cos(ImA1A2);
    aCylXA2   = CylYA2 * sin(ImA1A2) + CylXA2 * cos(ImA1A2);
    aCylXB1   = CylYB1 * sin(ImA1A2) + CylXB1 * cos(ImA1A2);
    aCylXB2   = CylYB2 * sin(ImA1A2) + CylXB2 * cos(ImA1A2);
    aConeXP   = ConeYP * sin(ImA1A2) + ConeYP * cos(ImA1A2);
    aConeXC1  = ConeYC1 * sin(ImA1A2) + ConeYC1 * cos(ImA1A2);
    aConeXC2  = ConeYC2 * sin(ImA1A2) + ConeYC2 * cos(ImA1A2);
    aCylMaxXA = max(aCylXA1, aCylXA2);
    aCylMaxXB = max(aCylXB1, aCylXB2);
    aCylMaxX  = max(aCylMaxXA, aCylMaxXB);
    aConeMinXC = min(aConeXC1, aConeXC2);
    aConeMinX = min(aConeXP, aConeMinXC);
```

```

        if aCylMaxX  < aConeMinX     %'shadows' on aX axis don't overlap
            DoesCollide  = false;
        else                        %'shadows' on aX axis do overlap
            mA1B1   = atand((CylYA1 - CylYB1) / (CylXA1 - CylXB1));
            ImA1B1  = mA1B1 + 90;    %axis perpendicular to b
            bCylXA1   = CylYA1 * sin(ImA1B1) + CylXA1 * cos(ImA1B1);
            bCylXA2   = CylYA2 * sin(ImA1B1) + CylXA2 * cos(ImA1B1);
            bCylXB1   = CylYB1 * sin(ImA1B1) + CylXB1 * cos(ImA1B1);
            bCylXB2   = CylYB2 * sin(ImA1B1) + CylXB2 * cos(ImA1B1);
            bConeXP   = ConeYP * sin(ImA1B1) + ConeYP * cos(ImA1B1);
            bConeXC1  = ConeYC1 * sin(ImA1B1) + ConeYC1 * cos(ImA1B1);
            bConeXC2  = ConeYC2 * sin(ImA1B1) + ConeYC2 * cos(ImA1B1);
            bCylMaxXA  = max(bCylXA1, bCylXA2);
            bCylMaxXB  = max(bCylXB1, bCylXB2);
            bCylMaxX  = max(bCylMaxXA, bCylMaxXB);
            bConeMinXC  = min(bConeXC1, bConeXC2);
            bConeMinX  = min(bConeXP, bConeMinXC);

            if bCylMaxX  < bConeMinX     %'shadows' on bX axis don't overlap
                DoesCollide  = false;
            else                        %'shadows' on bX axis do overlap
                mPC1    = atand((ConeYP - ConeYC2) / (ConeXP - ConeXC2));
                ImPC1   = mPC1 + 90;     %axis perpendicular to c
                cCylXA1   = CylYA1 * sin(ImPC1) + CylXA1 * cos(ImPC1);
                cCylXA2   = CylYA2 * sin(ImPC1) + CylXA2 * cos(ImPC1);
                cCylXB1   = CylYB1 * sin(ImPC1) + CylXB1 * cos(ImPC1);
                cCylXB2   = CylYB2 * sin(ImPC1) + CylXB2 * cos(ImPC1);
                cConeXP   = ConeYP * sin(ImPC1) + ConeYP * cos(ImPC1);
                cConeXC1  = ConeYC1 * sin(ImPC1) + ConeYC1 * cos(ImPC1);
                cConeXC2  = ConeYC2 * sin(ImPC1) + ConeYC2 * cos(ImPC1);
                cCylMaxXA  = max(cCylXA1, cCylXA2);
                cCylMaxXB  = max(cCylXB1, cCylXB2);
                cCylMaxX  = max(cCylMaxXA, cCylMaxXB);
                cConeMinXC  = min(cConeXC1, cConeXC2);
                cConeMinX  = min(cConeXP, cConeMinXC);

                if cCylMaxX  < cConeMinX     %'shadows' on cX axis don't overlap
                    DoesCollide  = false;
                else                        %'shadows' on cX axis do overlap
                    mPC2    = atand((ConeYP - ConeYC2) / (ConeXP - ConeXC2));
                    ImPC2   = mPC2 + 90;     %axis perpendicular to d
                    dCylXA1   = CylYA1 * sin(ImPC2) + CylXA1 * cos(ImPC2);
                    dCylXA2   = CylYA2 * sin(ImPC2) + CylXA2 * cos(ImPC2);
                    dCylXB1   = CylYB1 * sin(ImPC2) + CylXB1 * cos(ImPC2);
                    dCylXB2   = CylYB2 * sin(ImPC2) + CylXB2 * cos(ImPC2);
                    dConeXP   = ConeYP * sin(ImPC2) + ConeYP * cos(ImPC2);
                    dConeXC1  = ConeYC1 * sin(ImPC2) + ConeYC1 * cos(ImPC2);
                    dConeXC2  = ConeYC2 * sin(ImPC2) + ConeYC2 * cos(ImPC2);
                    dCylMaxXA  = max(dCylXA1, dCylXA2);
                    dCylMaxXB  = max(dCylXB1, dCylXB2);
                    dCylMaxX  = max(dCylMaxXA, dCylMaxXB);
                    dConeMinXC  = min(dConeXC1, dConeXC2);
                    dConeMinX  = min(dConeXP, dConeMinXC);

                    if dCylMaxX  < dConeMinX     %'shadows' on dX axis don't overlap
```

```
                            DoesCollide = false;
                        else                        %'shadows' on dX axis do overlap
                            mC1C2  = atand((ConeYC1 - ConeYC2) / (ConeXC1 - ConeXC2));
                            ImC1C2 = mC1C2 + 90;    %axis perpendicular to e
                            eCylXA1   = CylYA1 * sin(ImC1C2) + CylXA1 * cos(ImC1C2);
                            eCylXA2   = CylYA2 * sin(ImC1C2) + CylXA2 * cos(ImC1C2);
                            eCylXB1   = CylYB1 * sin(ImC1C2) + CylXB1 * cos(ImC1C2);
                            eCylXB2   = CylYB2 * sin(ImC1C2) + CylXB2 * cos(ImC1C2);
                            eConeXP   = ConeYP * sin(ImC1C2) + ConeYP * cos(ImC1C2);
                            eConeXC1 = ConeYC1 * sin(ImC1C2) + ConeYC1 * cos(ImC1C2);
                            eConeXC2 = ConeYC2 * sin(ImC1C2) + ConeYC2 * cos(ImC1C2);
                            eCylMaxXA = max(eCylXA1, eCylXA2);
                            eCylMaxXB = max(eCylXB1, eCylXB2);
                            eCylMaxX = max(eCylMaxXA, eCylMaxXB);
                            eConeMinXC = min(eConeXC1, eConeXC2);
                            eConeMinX = min(eConeXP, eConeMinXC);

                            if eCylMaxX < eConeMinX     %'shadows' on eX axis don't overlap
                                DoesCollide = false;
                            else                        %'shadows' on eX axis do overlap
                                DoesCollide = true;
                                %Since all possible overlapings are looked at
                                %and one does overlap, the objects collide.
                            end
                        end
                    end
                end
            end
        end
    t = t + TimeIncrement;
end
t = t - TimeIncrement;
if DoesCollide == false         %The objects do not collide
    timeCollide = 0;
else                            %The objects do collide
    timeCollide = t;
end
```

Appendix E – RadarConvert.m

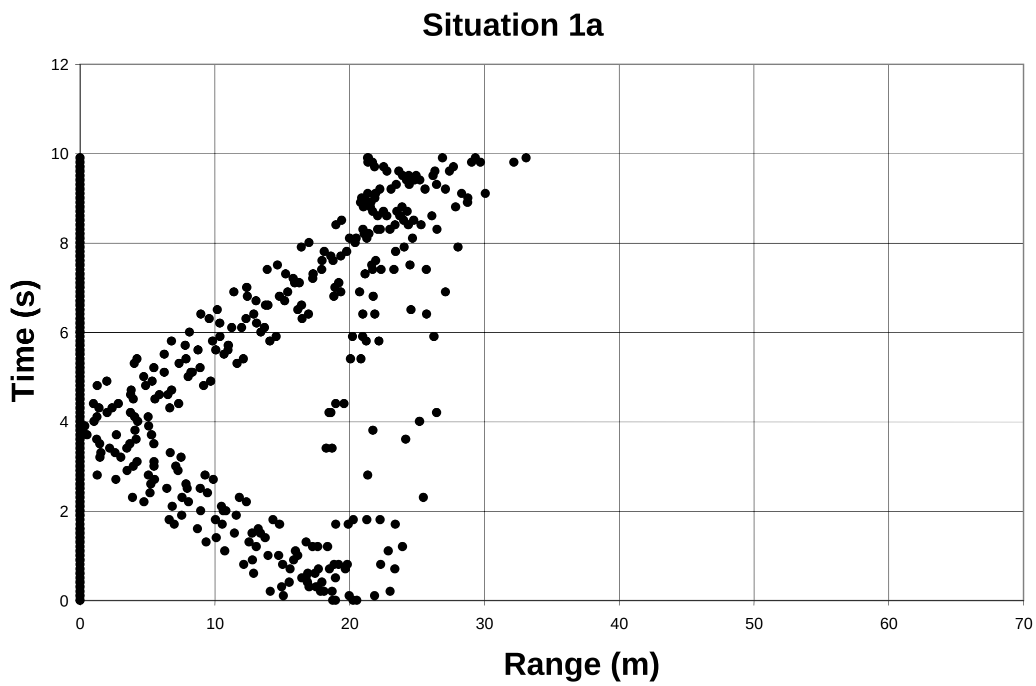
```
%RadarConvert.m
%WPI Physics MQP A-Term 2005
%MIT Lincoln Laboratory

function [A] = RadarConvert(CylCent, ConeCent, CylRot, ConeRot, CylLength, CylRadius, ConeLength, ConePointLength, CylBaseALength, t, X, Y);

%t                   Time
%CylCent             Velocity of cylinder
%ConeCent            Velocity of cone
%CylRot              Angular velocity of cylinder
%ConeRot             Angular velocity of cone
%CylLength           Length of cylinder
%CylBaseALength      Length from cylinder Base A to center of mass
%CylRadius           Radius of cylinder and cone base
%ConeLength          Length of cone
%ConePointLength     Distance from cone point to center of mass
CylBaseBLength = CylLength - CylBaseALength;
    %Distance from cylinder base B to center of mass
ConeBaseLength = ConeLength - ConePointLength;
    %Distance from cone base to center of mass

CylPosition = CylCent * t - CylBaseALength;
ConePosition = ConeCent * t + ConeBaseLength;
CylAngle = CylRot * t;
ConeAngle = ConeRot * t;

%   B1 ____________ A1   C1
%     |     b      |      |\ c
%     |           a|    e |  > P
%     |____________|      |/ d
%   B2              A2   C2

%Local X positions
CylXA    = CylPosition + CylBaseALength * cos(CylAngle);
CylXA1   = CylXA - CylRadius * sin(CylAngle);
CylXA2   = CylXA + CylRadius * sin(CylAngle);
CylXB    = CylPosition - CylBaseBLength * cos(CylAngle);
CylXB1   = CylXB - CylRadius * sin(CylAngle);
CylXB2   = CylXB + CylRadius * sin(CylAngle);
ConeXP   = ConePosition + ConePointLength * cos(ConeAngle);
ConeXC   = ConePosition - ConeBaseLength * cos(ConeAngle);
ConeXC1 = ConeXC + CylRadius * sin(ConeAngle);
ConeXC2 = ConeXC - CylRadius * sin(ConeAngle);

%Local Y positions
CylYA    = CylBaseALength * sin(CylAngle);
CylYA1   = CylYA + CylRadius * cos(CylAngle);
CylYA2   = CylYA - CylRadius * cos(CylAngle);
CylYB    = -1 * CylBaseBLength * sin(CylAngle);
CylYB1   = CylYB + CylRadius * cos(CylAngle);
CylYB2   = CylYB - CylRadius * cos(CylAngle);
ConeYP   = ConePointLength * sin(ConeAngle);
ConeYC   = -1 * ConeBaseLength * sin(ConeAngle);
ConeYC1 = ConeYC + CylRadius * cos(ConeAngle);
ConeYC2 = ConeYC - CylRadius * cos(ConeAngle);
```

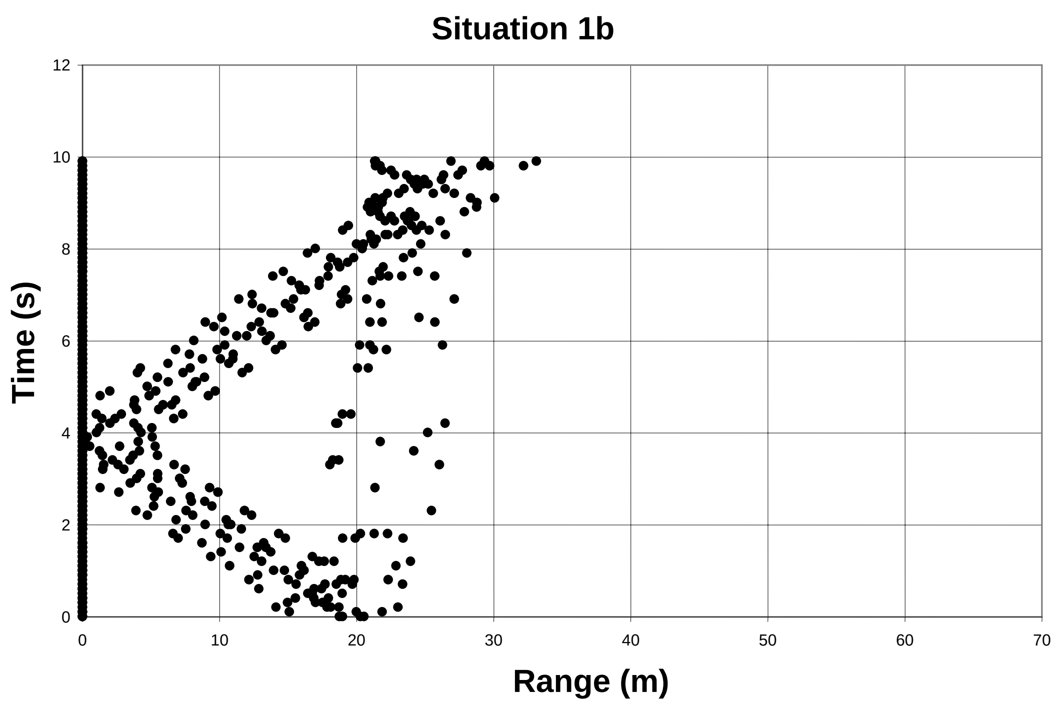
```
%Total X Positions
TCylXA1  = CylXA1 + X;
TCylXA2  = CylXA2 + X;
TCylXB1  = CylXB1 + X;
TCylXB2  = CylXB2 + X;
TConeXP  = ConeXP + X;
TConeXC1 = ConeXC1 + X;
TConeXC2 = ConeXC2 + X;

%Total Y Positions
TCylYA1  = CylYA1 + Y;
TCylYA2  = CylYA2 + Y;
TCylYB1  = CylYB1 + Y;
TCylYB2  = CylYB2 + Y;
TConeYP  = ConeYP + Y;
TConeYC1 = ConeYC1 + Y;
TConeYC2 = ConeYC2 + Y;

%Total Ranges from origin
RCylA1  = sqrt(TCylXA1^2 + TCylYA1^2);
RCylA2  = sqrt(TCylXA2^2 + TCylYA2^2);
RCylB1  = sqrt(TCylXB1^2 + TCylYB1^2);
RCylB2  = sqrt(TCylXB2^2 + TCylYB2^2);
RConeP  = sqrt(TConeXP^2 + TConeYP^2);
RConeC1 = sqrt(TConeXC1^2 + TConeYC1^2);
RConeC2 = sqrt(TConeXC2^2 + TConeYC2^2);
RCyl = [RCylA1 RCylA2 RCylB1 RCylB2] ;
RCone = [RConeP RConeC1 RConeC2] ;

%Total Angles from origin
AnCylA1  = atan(TCylYA1 / TCylXA1);
AnCylA2  = atan(TCylYA2 / TCylXA2);
AnCylB1  = atan(TCylYB1 / TCylXB1);
AnCylB2  = atan(TCylYB2 / TCylXB2);
AnConeP  = atan(TConeYP / TConeXP);
AnConeC1 = atan(TConeYC1 / TConeXC1);
AnConeC2 = atan(TConeYC2 / TConeXC2);
AnCyl = [AnCylA1 AnCylA2 AnCylB1 AnCylB2] ;
AnCone = [AnConeP AnConeC1 AnConeC2] ;

%Slopes of lines between points
SA1A2 = (TCylYA1 - TCylYA2) / (TCylXA1 - TCylXA2);
SA1B1 = (TCylYA1 - TCylYB1) / (TCylXA1 - TCylXB1);
SA1B2 = (TCylYA1 - TCylYB2) / (TCylXA1 - TCylXB2);
SA2B1 = (TCylYA2 - TCylYB1) / (TCylXA2 - TCylXB1);
SA2B2 = (TCylYA2 - TCylYB2) / (TCylXA2 - TCylXB2);
SB1B2 = (TCylYB1 - TCylYB2) / (TCylXB1 - TCylXB2);
SPC1 = (TConeYP - TConeYC1) / (TConeXP - TConeXC1);
SPC2 = (TConeYP - TConeYC2) / (TConeXP - TConeXC2);
SC1C2 = (TConeYC1 - TConeYC2) / (TConeXC1 - TConeXC2);


%Are points hidden by own body?

%A1
```

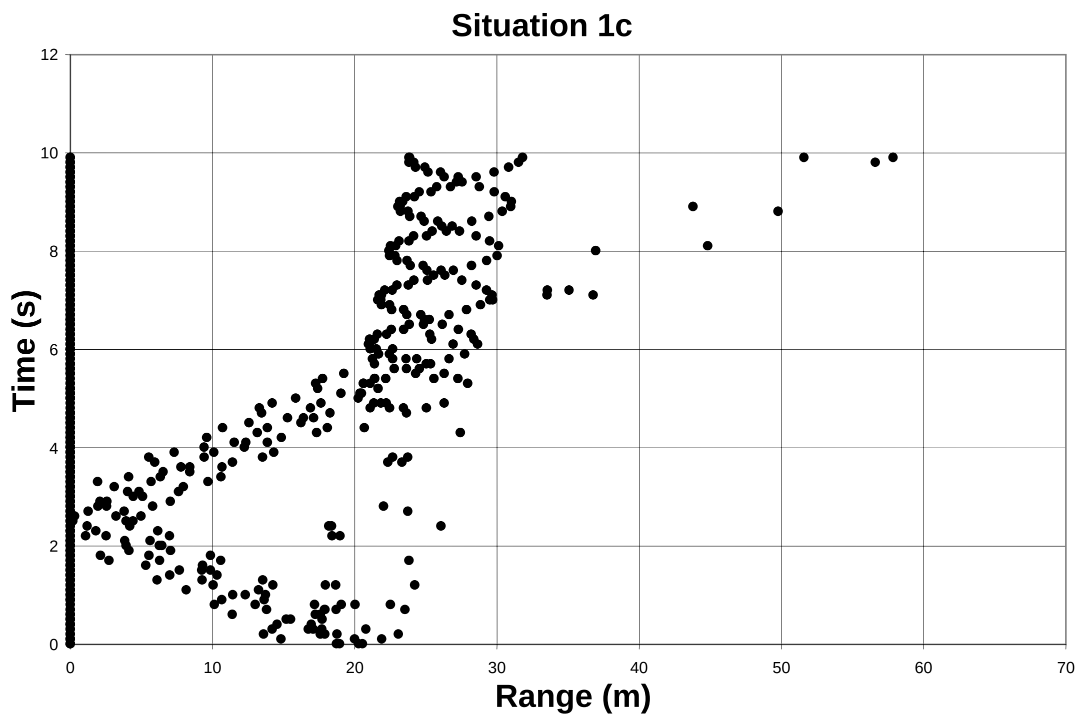
```
if (AnCylA1 <= AnCylB1) & (AnCylA1 >= AnCylA2)
    VisOwnA1 = false;
else
    VisOwnA1 = true;
end
%A2
if (AnCylA2 <= AnCylA1) & (AnCylA2 >= AnCylB2)
    VisOwnA2 = false;
else
    VisOwnA2 = true;
end
%B1
if (AnCylB1 <= AnCylB2) & (AnCylB1 >= AnCylA1)
    VisOwnB1 = false;
else
    VisOwnB1 = true;
end
%B2
if (AnCylB2 <= AnCylA2) & (AnCylB2 >= AnCylB1)
    VisOwnB2 = false;
else
    VisOwnB2 = true;
end
%P
if (AnConeP <= AnConeC1) & (AnConeP >= AnConeC2)
    VisOwnP = false;
else
    VisOwnP = true;
end
%C1
if (AnConeC1 <= AnConeC2) & (AnConeC1 >= AnConeP)
    VisOwnC1 = false;
else
    VisOwnC1 = true;
end
%C2
if (AnConeC2 <= AnConeP) & (AnConeC2 >= AnConeC1)
    VisOwnC2 = false;
else
    VisOwnC2 = true;
end

%Are points hidden by other body?

%the minimum and maximum angles for each shape and thier indices
[minCylAngle, minCylAnInd] = min(AnCyl);
[MaxCylAngle, MaxCylAnInd] = max(AnCyl);
[minConeAngle, minConeAnInd] = min(AnCone);
[MaxConeAngle, MaxConeAnInd] = max(AnCone);

%the ranges of the min and max angles for each shape
RmCylAn = RCyl(minCylAnInd);
RMCylAn = RCyl(MaxCylAnInd);
RmConeAn = RCone(minConeAnInd);
RMConeAn = RCone(MaxConeAnInd);

```

```
if (minCylAnInd == 1)
    if (MaxCylAnInd == 2)
        CylSlope = SA1A2;
    elseif (MaxCylAnInd == 3)
        CylSlope = SA1B1;
    else
        CylSlope = SA1B2;
    end
elseif (minCylAnInd == 2)
    if (MaxCylAnInd == 1)
        CylSlope = SA1A2;
    elseif (MaxCylAnInd == 3)
        CylSlope = SA2B1;
    else
        CylSlope = SA2B2;
    end
elseif (minCylAnInd == 3)
    if (MaxCylAnInd == 1)
        CylSlope = SA1B1;
    elseif (MaxCylAnInd == 2)
        CylSlope = SA2B1;
    else
        CylSlope = SB1B2;
    end
else
    if(MaxCylAnInd == 1)
        CylSlope = SA1B2;
    elseif (MaxCylAnInd == 2)
        CylSlope = SA2B2;
    else
        CylSlope = SB1B2;
    end
end

if (minConeAnInd == 1)
    if (MaxConeAnInd == 2)
        ConeSlope = SPC1;
    else
        ConeSlope = SPC2;
    end
elseif (minConeAnInd == 2)
    if (MaxConeAnInd == 1)
        ConeSlope = SPC1;
    else
        ConeSlope = SC1C2;
    end
else
    if (MaxConeAnInd == 1)
        ConeSlope = SPC2;
    else
        ConeSlope = SC1C2;
    end
end


%A1
```

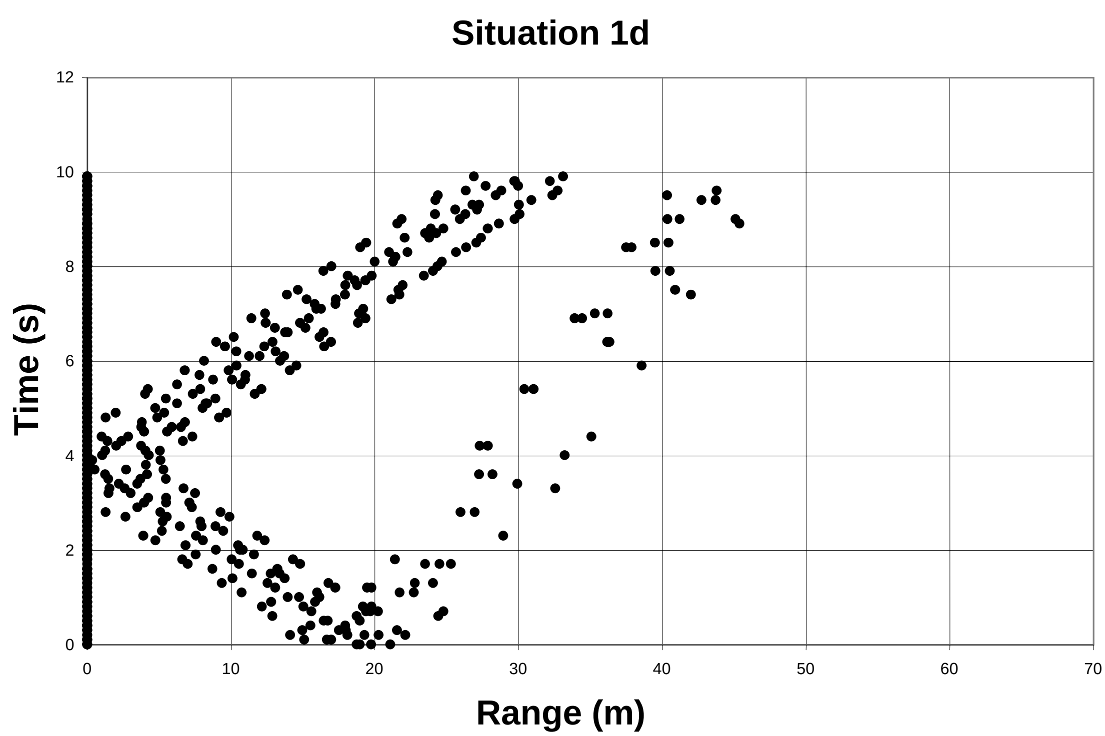
Appendix F – RTI Plots



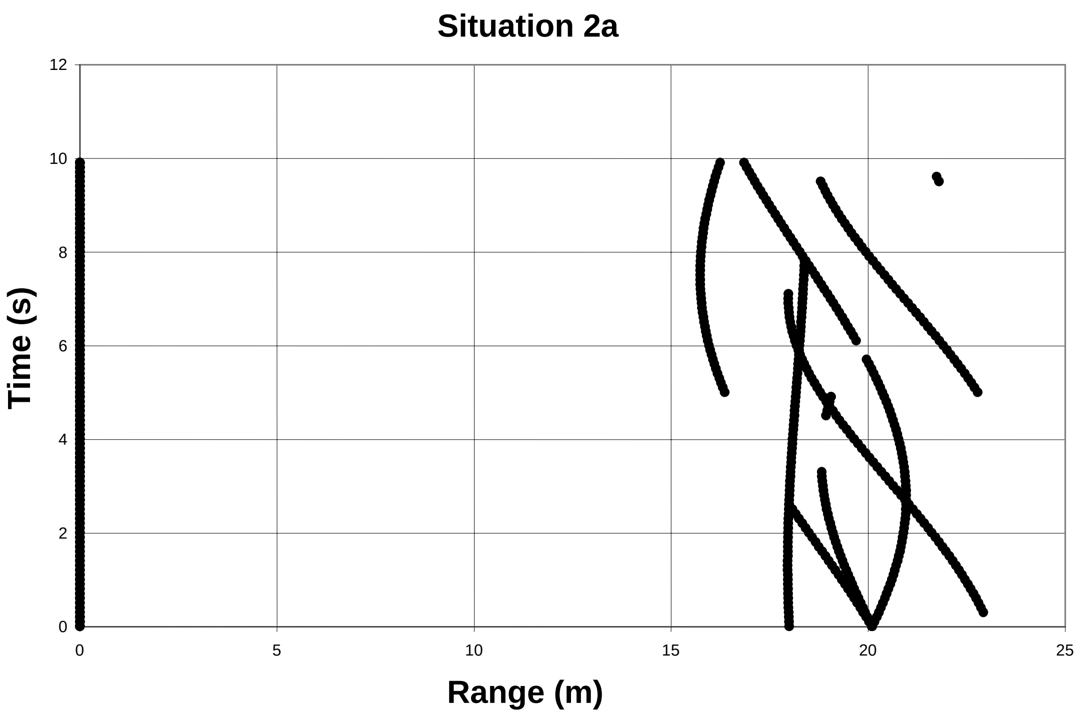




Situation 1c
Time (s)
Range (m)
0
2
4
6
8
10
12
0
10
20
30
40
50
60
70

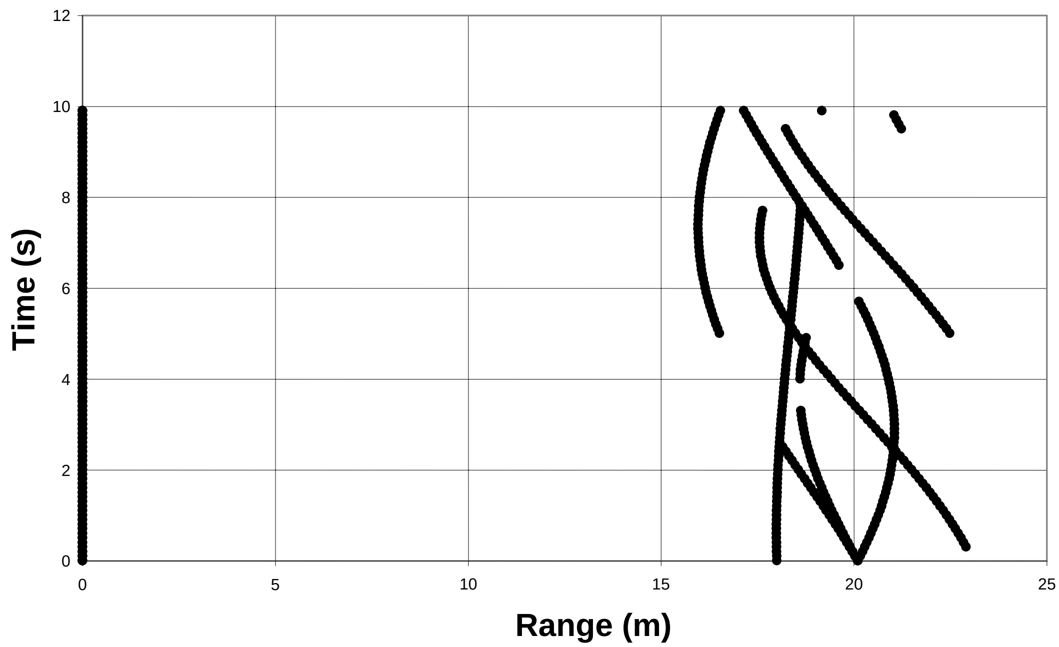


Situation 1d
Time (s)
Range (m)
0
2
4
6
8
10
12
0
10
20
30
40
50
60
70

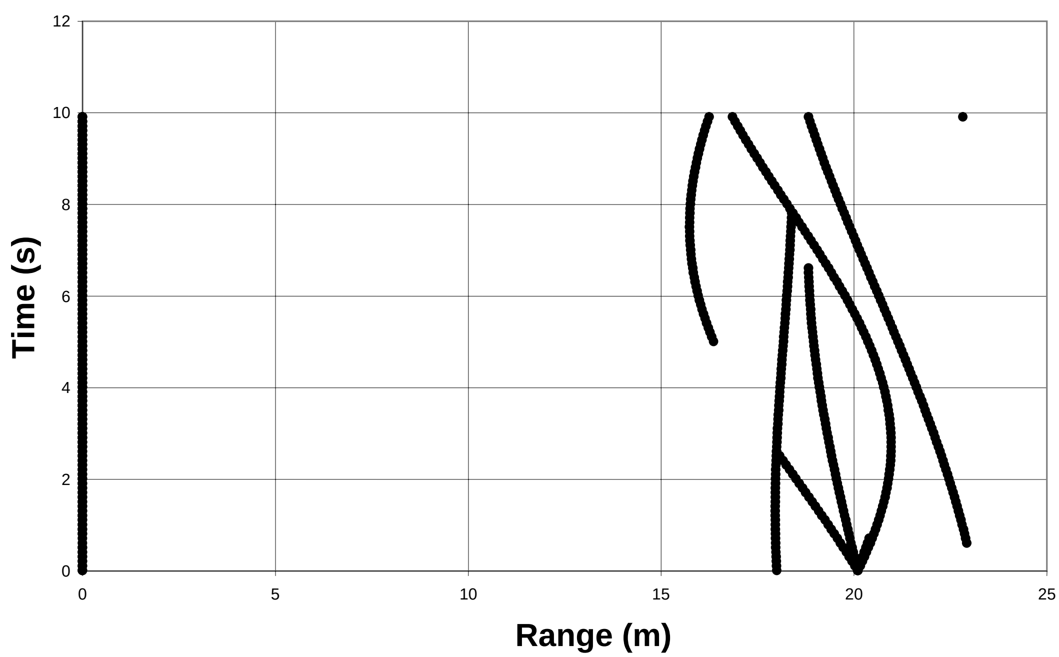

Situation 2a
12
10
8
6
4
2
0
0
5
10
15
20
25
Time (s)
Range (m)


Situation 2b
12
10
8
6
4
2
0
0
5
10
15
20
25
Time (s)
Range (m)

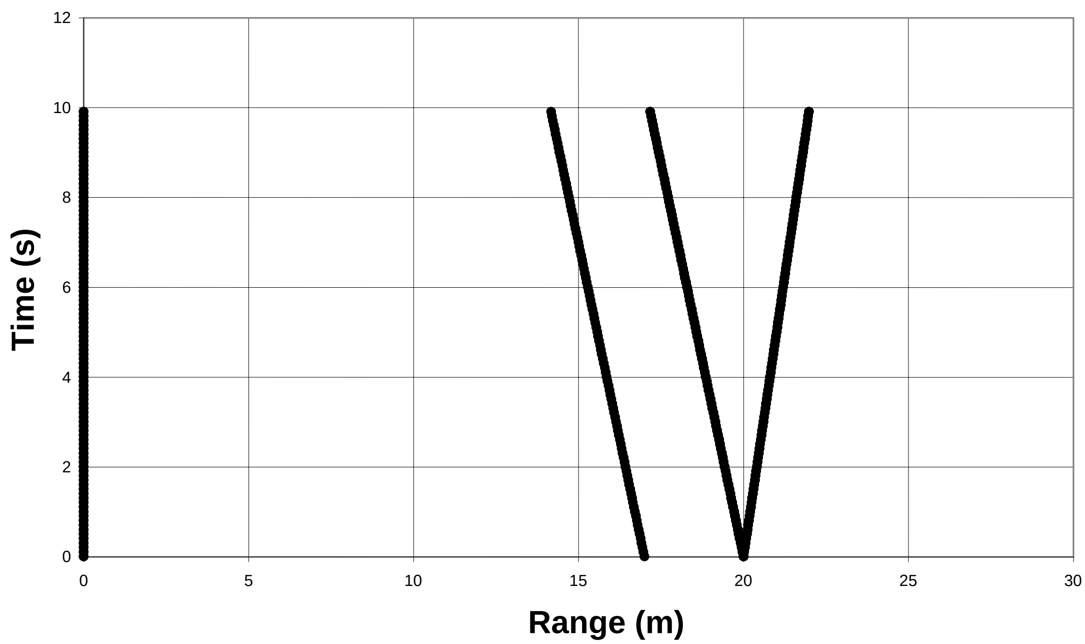
Situation 2c



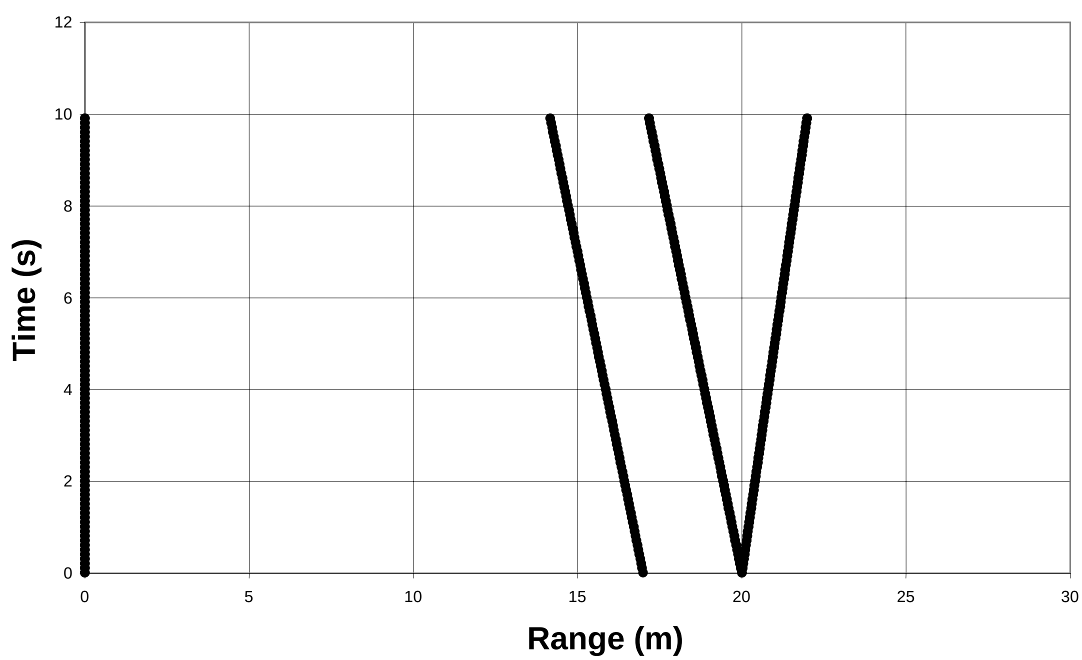
Situation 2d



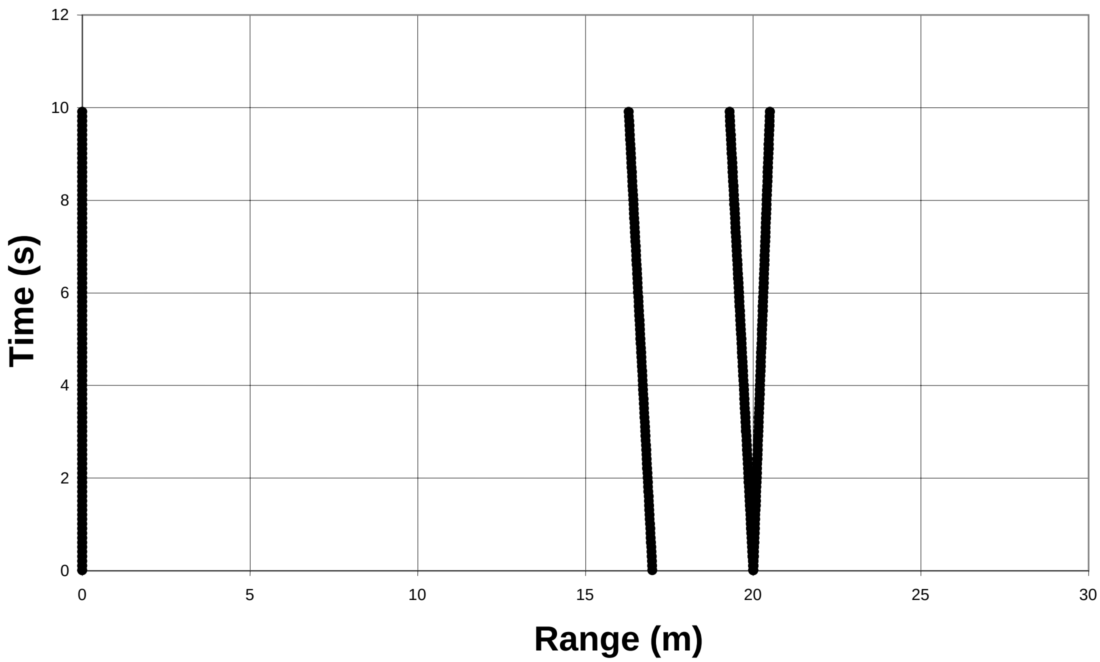
Situation 3a



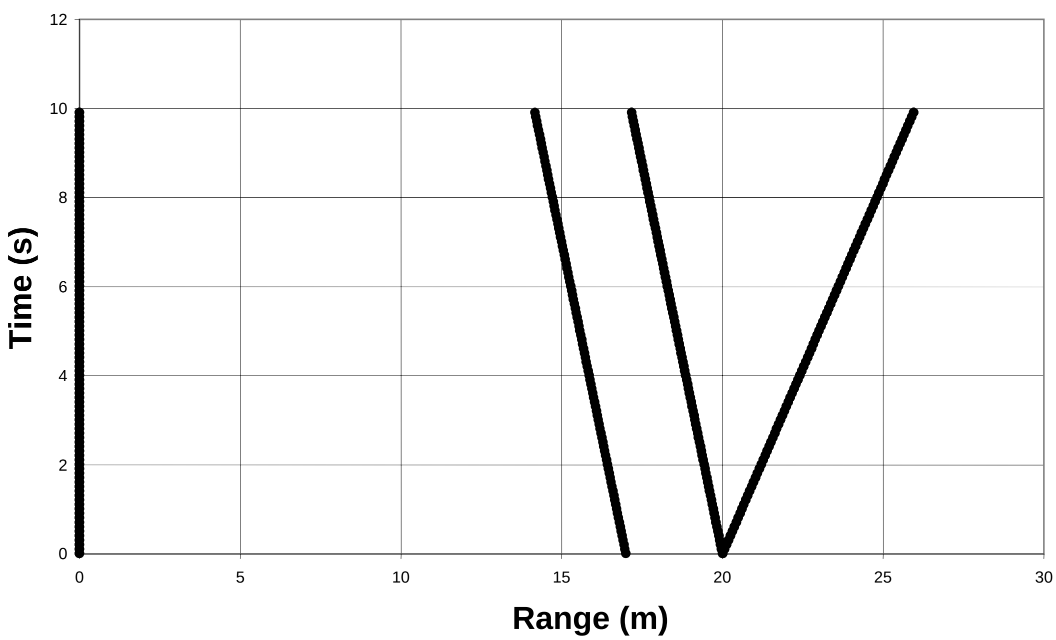
Situation 3b

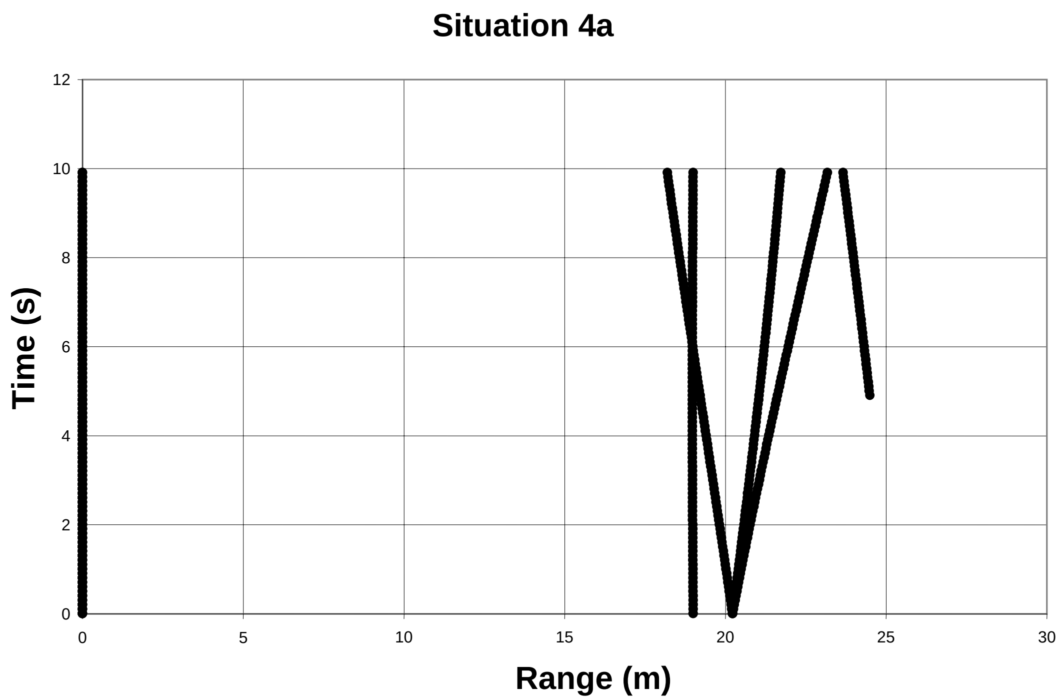


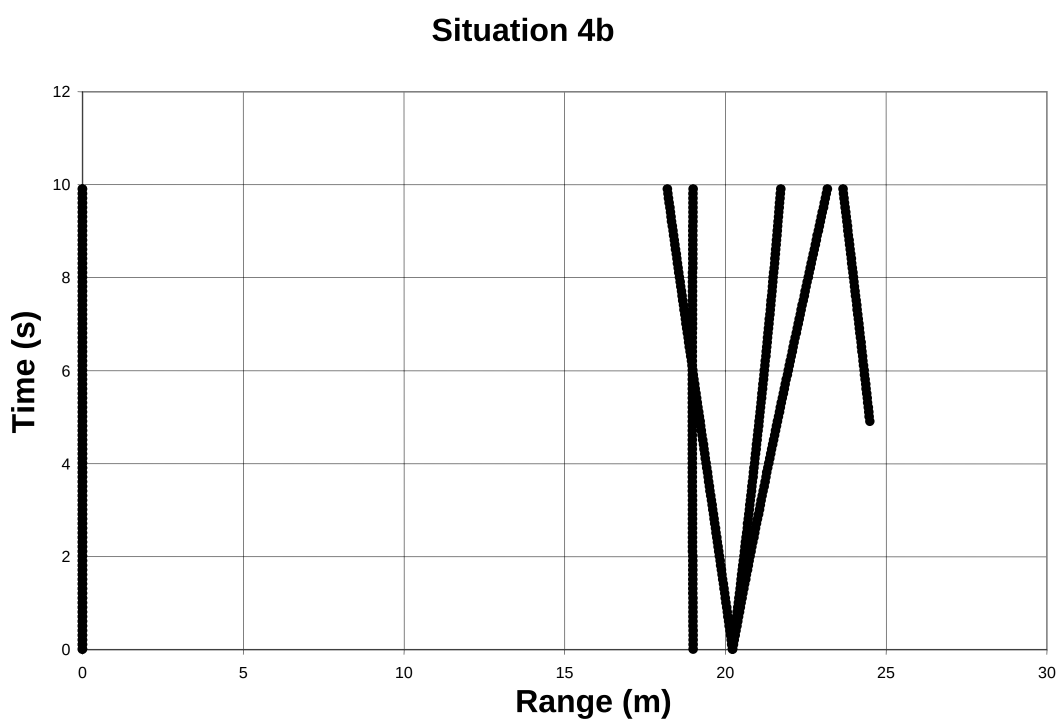
Situation 3c



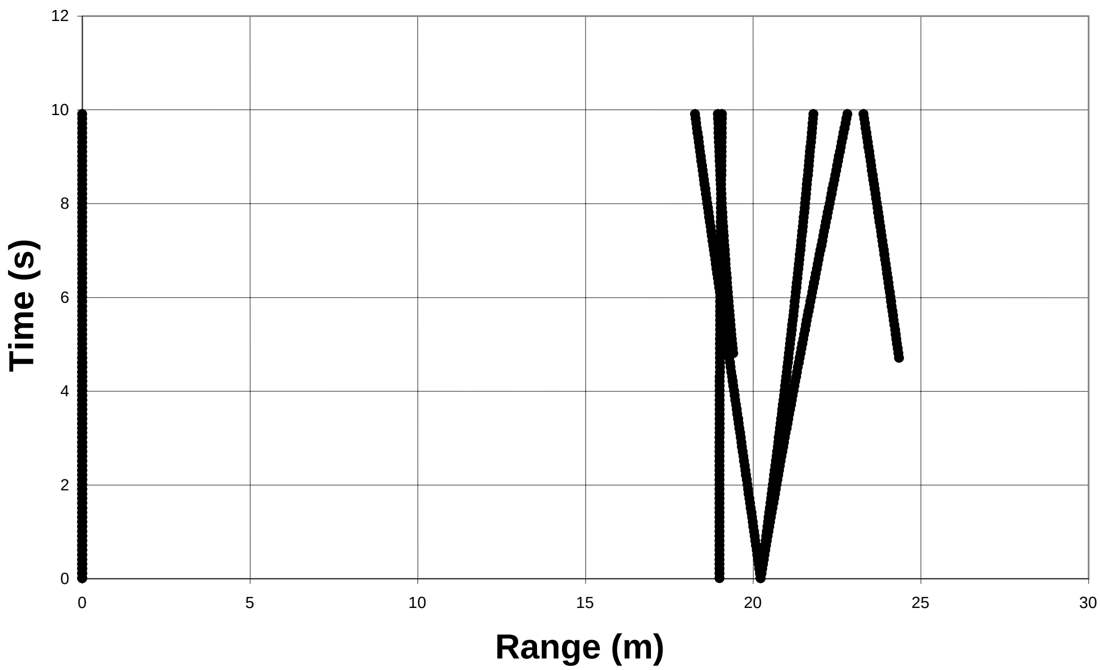
Situation 3d



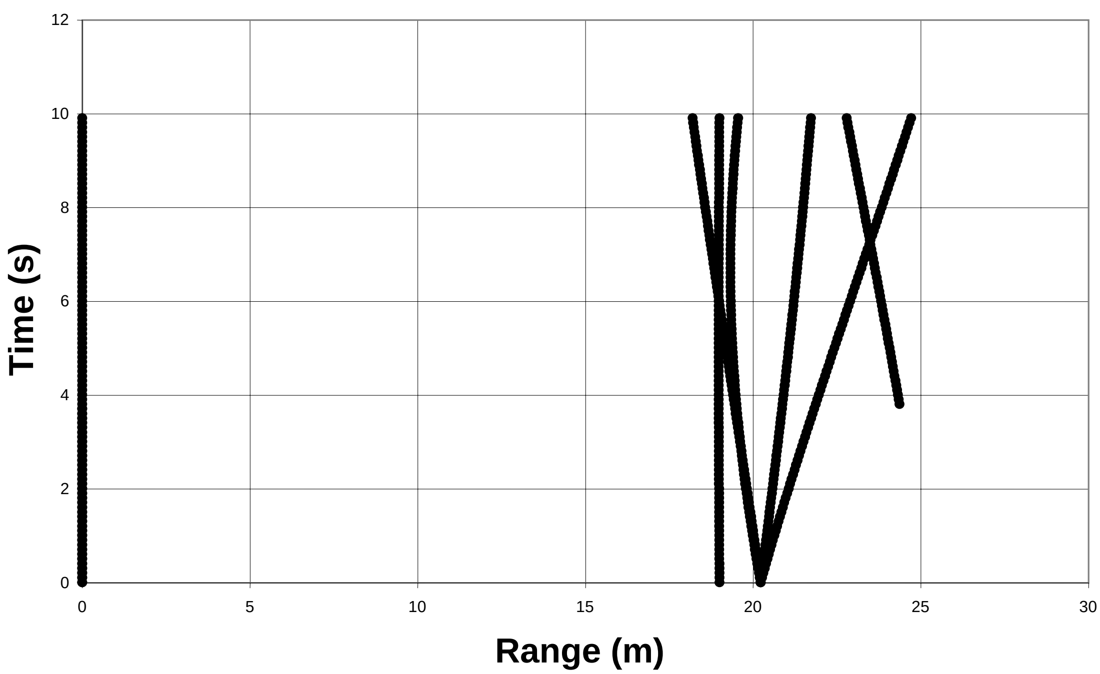

Situation 4a
12
10
8
6
4
2
0
Time (s)
0
5
10
15
20
25
30
Range (m)


Situation 4b
12
10
8
6
4
2
0
Time (s)
0
5
10
15
20
25
30
Range (m)

Situation 4c



Situation 4d



Bibliography

Edde, Byron. (1995). *Radar: Principles, Technology, Applications.* Upper Saddle River, NJ: Prentice Hall.

Ericson, Christer. (2005). *Real Time Collision Detection*. In Eberly, David H. (Series Ed) *Series in Interactive 3D Technology.* Amsterdam: Elsevier.

Skolnik, Merrill I. (2001). *Introduction to Radar Systems* (3rd ed). Boston: McGraw-Hill.

Wehner, Donald R. (1987). *High Resolution Radar*. Norwood, MA: Artech House, 1987.